University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations

1997

Scheduling in a dual resource constrained system using genetic algorithms.

Vishvas. Patel
University of Windsor

Follow this and additional works at: http://scholar.uwindsor.ca/etd

Recommended Citation

Patel, Vishvas., "Scheduling in a dual resource constrained system using genetic algorithms." (1997). *Electronic Theses and Dissertations.* Paper 3379.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

SCHEDULING IN A DUAL RESOURCE CONSTRAINED SYSTEM USING GENETIC ALGORITHMS

By

Vishvas Patel

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the Program of Industrial and Manufacturing Systems
Engineering in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada
September 1997
© 1997, Vishvas Patel

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-30982-7

Canada

I hereby declare that I am the sole author of this thesis. I authorize the University of Windsor to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Vishvas Patel

I further authorize the University of Windsor to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Vishvas Patel

The University of Windsor requires the signature of all persons using or photocopying this thesis. Please sign below, and give an address and a date.

ABSTRACT

Manufacturing organizations frequently face the challenges of making high-quality products quickly and of delivering those products to their customers on time. Improvements in the scheduling of their operations can often contribute to their success in meeting these goals. One criticism concerning scheduling research is that, too often the problems solved are far removed from reality. Most of the past research in the area of scheduling have considered the dispatching aspects of single-constrained shops. i.e. machine limited models. The machine limited model assumes that machines are idle only when there are no jobs waiting to be processed. On the other hand, it is required that both the machine and worker resources be available to process a job.

In this research a solution methodology and a software program system for assignment of dual resources (Machines and Workers) were developed. By considering two dual resource constrained problems as the candidate problems, an optimized GA based heuristic was developed. First genetic algorithms were used to determine the optimal staffing level which can be viewed as a basic design decision. The results show that for scheduling discrete manufacturing open shop with dual resources, where each worker has skill to operate two machines (flexibility level 2), and where the operator busy time is considered the same as the machine busy time, the optimal staffing level is 70%. Under these conditions, increasing the staffing level beyond 70% results in a minimal gain in the performance measures. Second, genetic algorithms were used to make short range control decisions regarding the operations of dual resource constrained shop. Six different

dispatching rules such as SPT, EOPNDD, EDD, FCFS, LSO, LPT and eight performance criteria were used. The performance of each rule with respect to each performance criterion for dual resource constrained shops is compared to single resource constrained shop. The results show that the rule which works best for a single resource constrained shop is not necessarily the best rule for a dual resource constrained shop. For the example considered SPT (Shortest Processing Time) rule performed fairly well for a DRC, while EOPNDD (Earliest Operation Due Date) rule performed well for a single resource constrained shop.

Finally, a system has been developed in 'C'. Scheduling data required for schedule optimization is inputted into the system by user interface. Genetic algorithms optimize the assignment of dual resources to each task. The genetic algorithms output a list of several assignment of dual resources to each task. This assignment of dual resources is then analyzed by the output analyzer which outputs a list of different performance criterion values, machine utilizations, worker utilizations, and ready & completion times of each order.

DEDICATION

To my Family for their
Love and Affection

ACKNOWLEDGMENTS

I would like to take this golden opportunity to acknowledge my deep sense of gratitude to Dr. Hoda ElMaraghy for supervising my research. Her vast knowledge, support, encouragement and helpful advice has been very instrumental in my research. It has been a great pleasure to be associated with her.

I would like to thank my committee members Dr. R. J. Caron and Dr. S. M. Taboun for their useful comments and helpful advice to improve the quality of work during the course of my program.

I would also like to express my sincere thanks to Dr. Ajay Jain for guiding me in the area of genetic algorithms. Special thanks to Jacquie Mummery and Penny Lappan for their daily assistance and support. I would also like to thank friends and colleagues of Intelligent Manufacturing Systems Centre for their invaluable help and support.

I gratefully appreciate the Department of Industrial and Manufacturing Systems Engineering for providing me the opportunity to be a teaching assistant. The financial support in the form of a research scholarship from Dr. Hoda ElMaraghy is also greatly appreciated.

Last but not the least I would like to express my deep appreciation to my parents, sisters and my wife Mina, for their constant support and encouragement throughout the work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

The following notations will be used in this research:

(i) i = Job Number i (i = 1, 2, ..., n)

(ii) k = Machine k (k = 1, 2, ..., m)

(iii) l = Worker l (l = 1, 2, ..., p)

(iv) t = Time

(v) J_i = Number of Operations of Job i

(vi) D_i = Order Size of Part i (i = 1, 2, ..., n)

(vii) M_i = Number of Machine Choice for each operation of a Part (i = 1 to n)

(viii) W_k = Number of Worker Choice for each Machine (k = 1 to m)

(ix) T_i = Processing Time of each Part (sum of processing times of each operation)

(x) O_{ij} = Operation Number j of job i

(xi) r_{ij} = Ready time of Operation j of job i

(xii) S_{ij} = Starting time of Operation O_{ij}

(xiii) C_{ij} = Completion time of Operation O_{ij}

(xiv) t_{ijk} = Processing time of Operation O_{ij}

(xv) E_{lk} = Efficiency of Worker l on Machine k

(xvi) d_i = due date for job i (i = 1,2, ..., n)

(xvii) U_k = Utilization for machine k (k = 1,2, ..., m)

(xviii) U_l = Utilization for worker l (l = 1,2, ..., p)

(xix) a_k = Release time for machine k

(xx) b_l = Release time for worker l

CHAPTER I

INTRODUCTION

1.1 General Introduction

Scheduling the work on a production floor has always been an important part of managing operations. It is very important to know which task has to be performed on which machine and by which worker. Management would like workers to perform the task in an order that helps the company to attains its production goals. The goals may include: completing orders on time, minimizing inventory, maximizing machine efficiencies, etc. The choice of scheduling rules, assignment of resources and methodology has a significant effect on whether or not these goals are met. Most manufacturing settings are quite complicated and determining a good schedule for these systems is difficult. Scheduling is considered to be the most difficult problem in manufacturing because of its NP-completeness, the algorithms in this class mostly have an exponential time i.e. the computational time increases exponentially with increase in problem size. Over the years, researchers have tried to solve this problem optimally and have discovered that optimal analysis of scheduling is very difficult in practice. The growing interest in genetic algorithms has led some production researchers to advocate these methods to manufacturing scheduling problems.

The efficient operation of a manufacturing system depends upon its physical components as well as the effectiveness of the planning and scheduling procedures. Production planning problems are concerned with those decisions that have to be made before the production system can actually begin to produce parts. Production Scheduling problems are concerned with the decisions regarding the running of the manufacturing system once it is set up during the planning stage.

1.2 Production Planning

In general, the term 'planning' is not well defined in literature, and is being used to describe a broad range of actions. In other words planning is a bridge between what we are and what we want to be. Taking a global view of factory organization, planning is done at three different levels (Verbraeck, 1991). At the top level, strategic planning is performed. Generally speaking it defines what to do. The result is a business plan which identifies company objectives and product groups. The company goals are taken as a guideline for the next lower level, the tactical one. Here resource planning is done. Based on the business plan elaborated at the strategic level, the resource plan allocates investments, operators and supply resources and determines the stocklevels. The task at the tactical level is to decide how and where to do it. At operational level, decisions are converted into a sequence of actions which specify when to do it.

It is important to note that upper level planning acts as a constraint on lower levels, but lower level planning has to provide the feed back which alone can keep the high level aims realistic. Also a strategic plan is concerned with the whole organization, whereas a detailed production plan affects only a small part such as shop floor activities.

1.3 Scheduling

Scheduling may be defined as the allocation of resources over time to perform certain tasks. It is a decision making process that has as a goal the optimization of one or more objectives. The resources and tasks may take many forms. The resources may be machines, workers, etc. in a work shop; runways at an airport; crews at a construction site; and so on. The task may be operations in a production process, take-offs and landing at airport, stages in a construction project and so on. The objective may also take many forms. One possible objective is minimizing of the mean flow time, and another is the minimization of the number of tasks completed after the committed due dates.

To produce a given part may require several manufacturing operations, according to the process plan. Each operation typically calls for the utilization of some machines and workers for some specific period of time. The first decision that must be made concerns the assignment of machines and workers to accomplish each operation on each scheduled part. The next decision is regarding the order in which the parts should be processed. These two decisions are called assignment and sequencing. Conway et. al (1967) and Baker (1974) have classified scheduling into subclasses of allocation (deciding which resources will be allocated to perform each task) and sequencing (deciding when each task will be performed and in which order). Thus scheduling is the final stage in production planning, the stage at which all the production activities are co-ordinated and projected on a time scale.

1.3.1 Basic Definitions

- *Task, Operation, Activity*: The basic items of work to be performed.
- *Resource, Machine, Worker*: Potential agent for performing work.
- *Jobs*: Sets of tasks grouped on the basis of characteristics they share in common.
- *Worker Flexibility*: The term used to indicate the relative ease with which workers can be shared between machines. A worker flexibility equal to 2 means, the worker can be shared between two machines.
- *Static Problem*: It assumes a predetermined, finite number of jobs, each with its own characteristics. Static problems are most often considered to be fully deterministic, all parameters such as processing times are assumed to be known constants either by definition or by accurate estimation.
- *Dynamic Problem*: It assumes an indefinite stream of jobs arriving for processing. They may be all different and arrive at random times, either singly or in batches. These problems are naturally stochastic, since jobs are constantly arriving and their characteristics are not known in advance, we may assume instead a probability distribution for them, based on past experience.
- *Lateness*: The deviation between a task's completion time and its due date. A task will have positive lateness if it is completed after its due date and negative lateness if completed before its due date.
- *Tardiness*: The measure of positive lateness. If a task is early, it has negative lateness but zero tardiness. If a task has positive lateness, it has equal positive tardiness.
- *Slack*: A measure of the difference between the remaining time to a task's due date and its processing time.

Much of the knowledge of scheduling policies is based on the static assumption. while it may seem that the real world is never static, infact practical operations often function in this mode; arriving orders are collected, and periodically a batch of jobs is scheduled. We assume static and deterministic settings hereafter.

1.3.2 Classification of Scheduling Problems

The Production Scheduling problem can be categorized according to the following schemes (Jain, 1995):

(a) *Requirements generation* , which distinguishes the manufacturing system as open or closed shop. In an open shop, all production orders are generated by customer requests and no inventory is stocked. In a closed shop, all customer requests are serviced from inventory.

(b) *Processing complexity*, which is concerned with the number of processing steps associated with each production task or item. A breakdown of this is : (i) Single Machine, Single stage (ii) Parallel Machine, Single stage (iii) Multistage, Flowshop, and (iv) Multistage, Job shop.

(c) *Data Variability*, which is concerned with the data involved i.e. whether the data are deterministic or stochastic.

(d) *Data time dependence*, whether the problem is static or dynamic. The problem is static if none of the initial data change overtime. It is considered dynamic if the data change with time.

1.4 Worker Scheduling

It has been long recognized that the shop performance is affected by the efficiency of the machines, productive work performed by the workers, and the manner in which the workers move between machines. The primary goal of worker scheduling is to have the right number of people available at the right time, at the same time satisfying the objective of minimizing direct worker cost. Although this objective is highly desirable, its achievement in manufacturing environments is highly dependent on the level of intelligence deployed in a particular system. Generally higher levels of intelligence require a more integrated system which, in turn, may require higher capital investment.

1.5 Problem Overview

The manufacturing scheduling problem has attracted a great deal of research and numerous approaches have been proposed. The development of good solutions to the manufacturing scheduling problem is confounded by two realities: the combinatorial complexity of the problem and the executional uncertainty of factory operations (Fox and Kempf, 1985).

From the past scheduling research it is observed that manufacturing scheduling is focused mainly on the problem optimization. Graves (1981) gives an extensive survey of the papers basically concerned with producing optimal solutions under various problem assumptions. Most researchers have simplified the scheduling problem in which dispatching rules have been studied by making an assumption that the only scarce resource is equipment, and worker is not treated as a constraining factor. However, inspection of many actual production facilities, especially job-shop or flow-shop configurations, reveals that the above assumption does not hold true.

1.5.1 Dual Resource Constrained System

A dual resource constrained (DRC) system is one in which shop capacity may be constrained by machine capacity, by worker capacity or by both. This situation exists in shops that have equipment which are not fully staffed and machine operators are capable of operating more than one piece of equipment. These operators may then be transferred from one machine to another, subject to skill restrictions.

The DRC system is much more difficult to model than the machine-limited system for two reasons. First, one must develop a rule that assigns available workers to machines as well as a dispatching rule to decide which job to process next at a given work station. Second, one must employ a specialized simulation program that has the capability of handling two constraining resources in this fashion. With respect to the problem of worker assignment, one could model a homogeneous worker resource, where all workers are equally efficient on each machine. One could also model a heterogeneous system, where on each machine the efficiency varies from worker to worker.

1.5.2 Problem Definition

The Dual Resource Constrained Scheduling problem consists, in a broad sense, of assigning, over the time, resources like machine and worker of finite capacity to operations, while complying with various constraints. Among the most important constraints one can find are as follows:

(a) *Capacity constraints* : at any time, not more than a certain number of resources of each type are available. Scheduling more operations on a type of resource that are available leads to an unfeasible schedule (e.g. it is impossible to perform three drilling operations simultaneously if there are only two drilling machines available).

(b) *Precedence constraints* : these are technological constraints, which they usually translate a necessary sequence of operations (e.g. first make a hole, then insert a screw).

(c) *Temporal constraints* : these usually translate the necessity to deliver the finished product before certain due date . However, the due date constraint is in most cases a soft constraint, meaning that it can be violated, but for the price of higher cost.

(d) *Multiflow Scheduling* : sometimes several kinds of resource are scarce, i.e. cannot be considered of infinite capacity. In those cases, several resources must be allocated for each operation, which increases the complexity of the scheduling problem. For instance, because worker payroll is one of the largest costs in many organizations, one could consider that workers are too expensive to be available in abundance.

(e) *Worker Flexibility* : sometimes workers are capable of operating more than one machine. In relation to this, two different situation exists, one of them is that a worker is usually occupied with the machine for loading, operation and unloading time. In this case no worker can process more than one job at a time. In second case the worker is usually occupied with the machine for loading and unloading time, this leads to the situation that worker is free during operation time and can work on different machine during that time interval.

The precedence constraints and the workers flexibility are the ones that make the Dual Resource Constrained Scheduling problem difficult to treat by genetic algorithms.

1.6 Objective

The objective of this study is to develop a solution methodology which makes good assignment of dual resources i.e. machine and worker, in order to attain specific production goals. The following steps are identified to achieve this objective:

(1) *Application of genetic algorithms to optimize the assignment of dual resources:* When alternate routing exists (routing flexibility) to process job orders and also workers are flexible (worker can work on more than one machine), vast number of assignment available to process the jobs in the manufacturing system, for this dispatching rules is employed in the genetic algorithms to obtain a satisfactory assignment of Dual Resources.

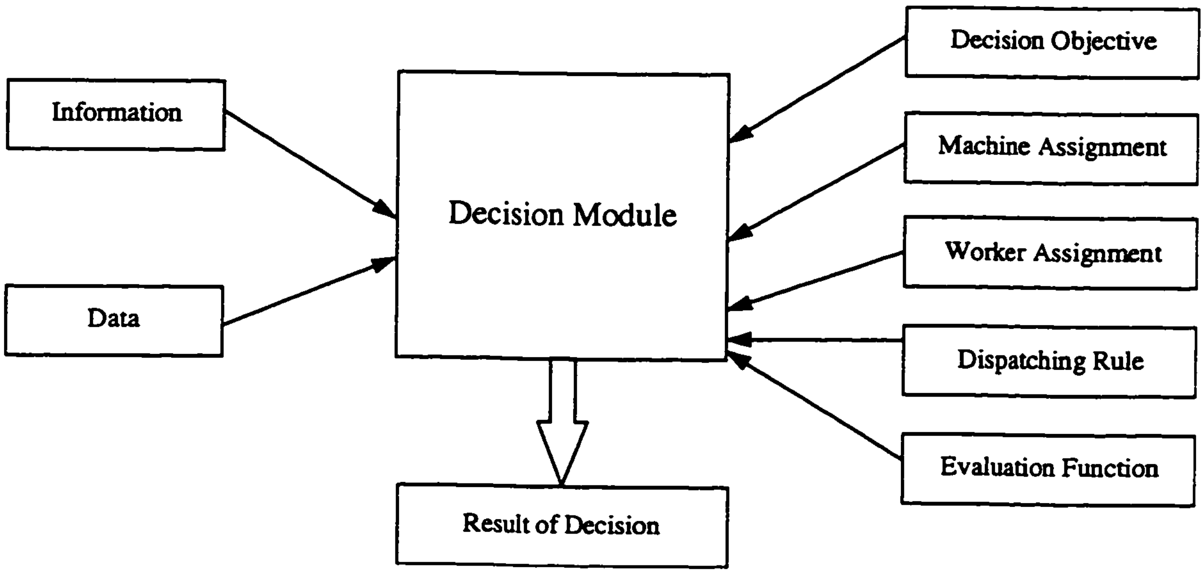


Figure 1.1 Basic Elements of Decision Net

(2) *System Implementation*: As a part of the system implementation a user interface is developed in C to input required Task/Machine information data and Worker/Machine information data. The final output is in the form of schedule indicating which task of which order has to be performed on which machine and by which worker, and start time and completion time of each task.

CHAPTER II

LITERATURE REVIEW

2.1 Scheduling Problem Issues

Manufacturing Scheduling has been an area of extensive and intense research over the past few decades. Research in manufacturing scheduling started primarily with simple analytical techniques, such as integer programming, and expanded to new solution approaches such as artificial intelligence and genetic algorithms. The vast amount of literature available to date on various scheduling problems have considered only equipment as the scarce resource and ignored the potential scarcity of workers.

The recent success of Japanese manufacturing has caused American manufacturers to search for ways in which it can regain some competitive advantage. One of the key elements of Japanese manufacturing, generally known as Just-In-Time (JIT) production, is a Group Technology (GT), where machines are grouped into manufacturing cells to produce families of parts with similar shapes or processing equipment's. The flexibility of a GT system is derived mostly from its allocation of flexible work forces. A flexible work force is achieved by worker cross-training that enables workers to perform a variety of tasks and relocate as the workload changes. The relative importance of sophisticated dispatching rules decreases with cross-trained workforces because the worker movement can reduce the occurrences of temporary bottleneck operations (Park, 1991).

The National Research Council (1991) highlights that manufacturing organizations are becoming flatter because of time-to-market pressures. This means that the responsibility is pushed to the shop floor and operators will be required to be more flexible to do more types of tasks that formerly were distributed among several workers with different job classifications. There have been several practical instances where worker flexibility has been able to improve quality while increasing manufacturing flexibility.

Most classes of scheduling problems fall into the categories of NP-hard problems (Garey and Johnson, 1979), which implies that for any algorithm which is guaranteed to find an optimal solution, the computational time grows exponentially with the size of the problem. Therefore, exact solution methodologies which can be implemented in polynomial time cannot be found for such problems. Many research efforts have focused on developing heuristics to provide good satisfactory, but not necessarily globally optimal solutions to these NP-hard problems. All solution approaches to various scheduling problems tried so far are mainly based on either simulation, operation research or artificial intelligence techniques. The search space in all these approaches grows exponentially with the problem size. For Dual Resource Constrained System with machines and workers as the limiting resources, the scheduling problem becomes more complicated as the skill or flexibility of the workers increases. As a result of this, the feasible number of assignment for machines and worker resources grows exponentially and finding a satisfactory assignment of resources becomes a formidable task.

The literature review is organized as follows:

(a) Literature survey pertaining to Dual Resource Constrained System

(b) Literature survey pertaining to the Genetic Algorithms approach to scheduling problems.

2.2 Dual Resource Constrained system

A Dual Resource Constrained (DRC) system is one in which all equipment in the shop is not fully staffed (Treleven, 1989) and further more, the operators can be transferred from one piece of equipment to another as needed. For Dual Resource Constrained System the cross-training of worker is very beneficial since it enables organizations to utilize their worker resources more efficiently (Allen, 1963). DRC literature is segregated into two classes: research concerned with operation of Dual Resource Constrained shop, which involves developing worker assignment rules; and research concerned with the design of DRC job-shop. In DRC systems decisions have to be made about when to consider transferring workers and given that they are eligible for transfers, to which areas these workers should be transferred (Treleven, 1989).

The Dual Resource Constrained Shop can be represented as follows:

(a) Worker Differential Scheme (LD): A Shop having workers of same efficiency at all machine centers.

Machine / Worker	1	2	3
1	1.0	1.0	1.0
2	0.9	0.9	0.9

Table 2.1 Worker Differential Scheme

(b) Machine Center Differential Scheme (MCD): A Shop with similar worker skill levels but machines of varying capabilities and efficiencies.

Machine / Worker	1	2	3
1	1.0	0.95	0.9
2	1.0	0.95	0.9

Table 2.2 Machine Center Differential Scheme

(c) Worker and Machine Center Differential (L&MCD): A Shop with different skilled workers and machines of varying capabilities and efficiencies.

Machine / Worker	1	2	3
1	1.0	0.95	0.9
2	0.75	1.0	0.95

Table 2.3 Worker and Machine Center Differential Scheme

The literature related to Dual Resource Constrained with machines and workers as the limiting resources can be classified along with two important shop design dimensions:

(a) A shop consisting of Homogeneous Resources

(b) A shop consisting of Heterogeneous Resources

2.2.1 Homogeneous Resources

Homogeneous resources are similar resources such as

Machine / Worker	1	2	3
1	1.0	1.0	1.0
2	1.0	1.0	1.0

Table 2.4 Homogeneous Resources

Decision rules to determine where to assign workers has minimal impact on shop performance, for the shop consisting of homogeneous worker resources (Fryer 1979, Treleven and Elvers 1985, Weeks and Fryer 1976). Decision rules to determine when to transfer a worker have a significant impact on the shop performance (Fryer 1974, Fryer 1976, Gunther 1979, Rochette and Sadowski 1976, Treleven 1987).

2.2.2 Heterogeneous Resources

Heterogeneous resources are dissimilar resources, here the workers are considered to be a heterogeneous resource. From the table 2.5, it can be observed that workers are having different efficiencies for different machine, for example worker 1 is able to operate machine 1 with 100% efficiency and worker 2 is able to operate machine 1 with 40% efficiency. Because of this different assignment of workers to same machine will results in different completion time to process a task.

Machine / Worker	1	2	3
1	1.0	0.7	0.4
2	0.4	0.7	1.0

Table 2.5 Heterogeneous Resources

Nelson (1970), investigated that with heterogeneous workers, system performance for the shop is sensitive to the type of 'When' rule chosen. The 'When' rule can be made under two different types of environment. The first is centralized worker assignment rule, under which workers are eligible for transfers after the completion of every job. The second is decentralized, under which worker transfers only when workers become idle on the machine to which they are assigned. Research under these two conditions indicate that for centralized worker assignment rule, the increased frequency of worker transfers results in a reduction of the time spent by workers at machines where they are inefficient, and thereby improving system performance at the cost of higher number of transfers. The decentralized rule was shown to worsen performance, with decentralized policy workers have a greater chance of being trapped in the departments in which they are inefficient, thereby further adding to the work backlog and reducing their chances of being transferred out of that department (Malhotra & Kher, 1994).

On the other hand, Gunther (1979, 1981) and Treleven (1987) recommended that for shops with transfer delays, a rule that delays transfer is preferable, since, increasing the transfers worsens system performance in such an environment.

2.2.3 Shop Size and Worker Flexibility

Fryer (1975), in his studies examined the influence of system size and worker flexibility on the performance effects of dispatching and worker control decision rules in Worker and Machine limited production systems. The result of his research indicates that with different size and worker flexibility characteristics, the performance changes for a given decision rule across all production systems. Small systems do act as reasonable predictors of large systems (Hogg, Phillips, & Maggard, 1977). Larger systems yield consistently better performance regardless of worker assignment rule and are preferable to separate small systems. The performance of small system is very sensitive to the worker assignment procedure, whereas the performance of large system is much less sensitive.

2.2.4 Job Routing Pattern

Another important parameter to be noted is the impact of job routing pattern on performance of dual resource constrained systems. The results of research done by (Treleven & Elvers, 1985) shows that the more flow-shop routings there are, the more effective the dispatching rules become in processing the work load.

2.2.5 Worker Flexibility and Staffing Levels

In a DRC environment, shop floor control decisions, in addition to well known dispatching and lot sizing decisions, include such decisions as staffing levels and worker assignments. Essentially, staffing level decisions determine the number of people allocated to each department while worker assignment determines which worker is allocated to which machine in the department. Worker flexibility of two and staffing level of 60-70% represents the best combination of the two strategies for better performance of DRC system with Homogeneous worker resources (Felan III, Fry, Philipoom, 1993).

Recent studies by Bobrowski & Park (1993) recognized a common and real element in the job shop environment, that is, all workers are not uniformly capable or experienced to operate all machines with the same degree of skill. In the environment where the worker is not perfectly interchangeable across the work centers, the selection of 'Where' rules should be made first considering the workers capability, followed by the choice of dispatching rule and then finally the selection of 'When' rule is used in descending order of system improvement that each factor contributes.

2.3 Genetic Algorithms

Genetic Algorithms (GA) were introduced by Holland (1975) as a method for modeling complex systems. Genetic Algorithms apply concepts from biological evolution to a mathematical context. The general idea is to start with randomly generated solutions and implement a "Survival-of-the-fittest" strategy to evolve good solutions. See Goldberg (1989); Davis (1991); Beasley, Bull & Martin (1993a); Khuri, Back & Heitkotter (1994) and Michalewicz (1994) for a detailed introduction to GA and some of their applications.

Genetic algorithms have been applied to scheduling problems by several authors. The primary strategy that has been used is a literal permutation ordering encoding [Bagchi et at. (1991); Biegal and Davern (1990); Cleveland & Smith (1989); Falkenauer & Bouffouix (1991); Gupta, Gupta, & Kumar (1993)]. Other encoding strategies include: a binary encoding based on prioritizing jobs by Nakano (1991) and Fox & McMahon (1991), an order based encoding presented by Grefenstette et al. (1985), and an indirect encoding introduced by Davis (1985).

Falkenauer and Bouffouix (1991) applied genetic algorithms on small, medium and large job shop problems. They introduced a suitable encoding of solutions of the problem together with a new crossover operator exploiting the encoding and a cost function to estimate the quality of solutions. The two crossover operators they used were PMX

(Partially Matched Crossover) and LOX (Linear Order Crossover). Their experiments reveal the superiority of the genetic algorithms over the common scheduling heuristics and a slight superiority of the LOX operator over the PMX operator.

Gupta et al. (1993) address an n-job, single machine problem with an objective to minimize the flow time variance. They proposed a heuristic procedure based on genetic algorithms with the potential to address more generalized objective function such as weighted flow time variance. They also used PMX operator for their experiments and mention investigating the performance of other crossover operators.

Nakano and Yamada (1991) used a conventional genetic algorithm to solve a job shop scheduling problem. They introduced three unique ideas in representation, evaluation and survival. They used binary representation which made it possible for them to use conventional genetic algorithms. Even though a chromosome produced by a conventional crossover is usually illegal, the evaluation method evaluates it by finding a similar legal chromosome. To help chromosomes survive, they introduced a new treatment, called forcing, that replaces illegal chromosomes with legal chromosome. It was shown that use of forcing improves convergence rates and solution quality.

Jain and ElMaraghy (1995) have applied genetic algorithms for scheduling in a batch manufacturing systems. They have introduced three new batch splitting policies. Their experiments reveal the superiority of new batch splitting policies namely, Split according to machine requirements, Split according to total number of tasks and Split according to processing times, over the batch splitting based on queuing models. First, appropriate batches were formed based on the batch splitting policies and then detailed scheduling of these batches was performed using genetic algorithms. The results of their research reveal that the use of appropriate batch policies improves the performance of the manufacturing systems.

Norman (1995) applied genetic algorithm for scheduling problems containing complexities such as multiple process plans, nonidentical machines, release times, due times, sequence dependent setup times, and have the total tardiness as the objective. The random keys representation was used for encoding a solution with random numbers. He concluded that random keys encoding may be better than the other encodings, the reason for this may be that it permits the use of the standard crossover operator during the genetic search.

Uckun, Bagchi, Kawamura and Miyabe (1991) addressed the problem of alternative process plans and developed two new chromosome representation where each chromosome represents a complete schedule. They have discussed three different representational schemes, namely direct, indirect and problem specific representation. To enhance the performance of the algorithm and to expand the search space, a chromosomes representation that stores problem specific information was devised and compared with the other two representations. The crossover operator used was PMX.

Murata, Ishibuchi and Tanaka (1996) have applied genetic algorithm to multi-objective optimization of a flow-shop scheduling problem. They have tested the algorithm with three objectives: to minimize the makespan, to minimize the total tardiness and to minimize the total flow time. The results of their experiments reveal that their multi-objective genetic algorithm could find better solutions than the single-objective genetic algorithm.

Chen, Neppalli & Aljaber (1996) generated a GA based heuristic for continuous flow shop problems with total flow time as the criterion. The results of their research provided three important points for applications of GA. First, the parameter values of GA are problem specific. Different parameters values of GA may yield significantly different results for the same problem. Second, some specific knowledge of the problem under consideration may improve the performance of GA. Third, completely randomly generating the initial population of GA may not be a good approach, specifically, for the larger size problem.

Jain and ElMaraghy (1997) have applied genetic algorithms techniques to the manufacturing scheduling problems in FMS. They have used two new operators for genetic recombination; reduced surrogate crossover, for crossing two parent schedules; and adaptive mutation, for random introduction of new genes in the chromosome (schedule). The results of their research reveal that as genetic algorithms performs multi-point search as opposed to a single point, the computational time required to reach an optimal or near optimal solutions was small compared to other approaches. Consequently genetic algorithms can be used to rescheduled the manufacturing tasks dynamically in case of interruptions like machine breakdown, cancellation of orders, arrival of rush orders and increased priority of a job. They have proposed rescheduling algorithms when uncertainties that cause discrepancies between the actual output and the planned output take place. Genetic algorithms were used to obtain an initial schedule. When an interruption occurs, the system status is updated and the genetic algorithms are rerun at that point of time to reschedule the remaining manufacturing tasks.

2.4 Motivation for the Proposed Research

Over the past few decades, a number of survey, classification, and review articles have focused on scheduling research in machine constrained job shops. It has been long recognized that shop floor management includes not only the scheduling of machines directly involved in production but also the scheduling of other needed resources like workers. The shop performance is affected by the efficiency of the machine/part, productive work performed by the worker, and the manner in which the operator is transferred between machines. Hence, worker scheduling is a very important issue. The primary goal of the worker scheduling is to have the right number of people available at the right time. While it is important to minimize payroll costs, it is important also to provide predictable, stable work schedules for workers, because that will keep morale up and turnover down. Because payroll is one of the largest cost in many organizations, a relatively minor improvement in scheduling jobs in a dual resource constrained environment can provide a significant return on investment.

The complexity of manufacturing systems makes it difficult to address all issues simultaneously. From past research in a Dual Resource Constrained system, it is observed that researchers have focused mainly on developing worker assignment rules using simulation techniques. Simulation approaches are costly in terms of computational time and the required human modeling efforts. It has been proved that genetic algorithms present a good solution methodology and provide better results than the heuristics for solving scheduling problems. The application of genetic algorithms to date suffer from major drawback, since the experiments considered only single-constrained job-shops, i.e.

machine-limited model. This research considers, a very important issue that has not yet been considered, which is shops constrained not only by machine resources but also by worker resources. The objective of this research is to determine the optimal work force required in the system, which may be viewed as a basic design decision as far as the initial hiring and on going assignments are concerned. This research also helps in making short-range control decision with respect to the allocation of a fixed work force among various components or machines of a system in response to the current system state.

CHAPTER III

GENETIC ALGORITHMS

3.1 Introduction

In the past, substantial time has been spent by researchers to develop practical and efficient methods for solving different types of scheduling problems. For most of the scheduling problems, the fundamental question is the number of iterations required to find an optimal solution. In their basic form, scheduling problems belong to a special class of `hard' problems, the so-called NP-hard or NP-complete problems for which no polynomial time algorithm has been found. The algorithms in this class mostly have an exponential time behavior. No fast solution method exists for NP-hard problems. This means that for large scheduling problems, no optimal solutions can be computed in a reasonable amount of time. Therefore, we have to be satisfied with a schedule that is not optimal but the best possible for a given situation. This means that the problem of scheduling is one of satisfying rather that optimizing.

Researchers are continuously making serious effort to reduce the gap between scheduling theory and practice, and as a result much of the previous research effort has been spent developing more powerful algorithms and more effective heuristics for manufacturing scheduling problems. Genetic algorithms are optimization techniques that have been applied successfully to many NP-hard problems. Genetic algorithms belong to a class of probabilistic algorithms, which combine elements of directed and stochastic searches. The important property of such genetic-based search methods is that they maintain a population of potential solutions, all other methods process a single point of the search space.

Most of the techniques used for optimization are iterative improvement techniques, where the technique is applied to a single point (the current point) in the search space. During a single iteration, a new point is selected from the neighborhood of the current point. If the new point provides a better value of the objective function, the new point becomes the current point; otherwise, some other neighbor is selected and tested against the current point. The method terminates if no further improvement is possible.

The genetic algorithms perform a multi-dimensional search by maintaining a population of potential solutions and encourages information creation and exchange between these directions. Using past information, they direct search with expected improved performance and achieve fairly consistent and reliable results (Whitley, 1987). The population undergoes a simulation evolution, i.e. at each generation the relatively good solutions reproduce , while the relatively bad solutions die. To distinguish between different solutions, an objective function (also called evaluation function or fitness function) is used which measures the quality of the solution. For more detailed discussion of the genetic algorithms and their use, the reader may refer to Goldberg (1989) and Michalewicz (1992).

3.2 Types of Genetic Algorithms

Genetic algorithms vary from practitioner to practitioner, but the basic building blocks for each genetic algorithms remains the same. There are two types of genetic algorithms currently in use, namely standard genetic algorithms (also known as traditional algorithms) and steady state genetic algorithms.

3.2.1 Standard Genetic Algorithms

These algorithms normally employ binary encodings, generational reproduction and simple one-point crossover and mutation operators. The population size remains unchanged during the evolution process, two mating parents create only two children but are themselves eliminated (Holland, 1975). The basic characteristics of these algorithms are as follows:

(a) Reproduction is based on fitness value and crossover rate. Strings are reproduced according to their fitness value and parent strings are selected according to crossover probability, $p_{C,}$ for off-spring production.

(b) Multiple reproduction and recombination occurs at one time.

(c) Parents die and are replaced by their children.

(d) The feedback is slower, because operators are applied on more than one pair of strings.

(e) Chances of duplicate strings being created are greatly increased.

The basic structure of standard genetic algorithms is shown in figure 3.1.

3.2.2 Steady State Genetic Algorithms

Standard genetic algorithms have several drawbacks. For standard genetic algorithms after the crossover application, parent strings on which the crossover was applied are lost, and all the parents sets are replaced by their children. Hence, the best solution found may be lost. A second drawback for standard genetic algorithm is, due to the nature of genetic operators. Chances of duplicate strings being created are greatly increased. These two drawbacks can be overcome by use of Steady-State Genetic Algorithms which was introduced by Whitley and Kauth (1988) in their genetic algorithm software package called `GENITOR'. The main characteristics of these algorithms are:

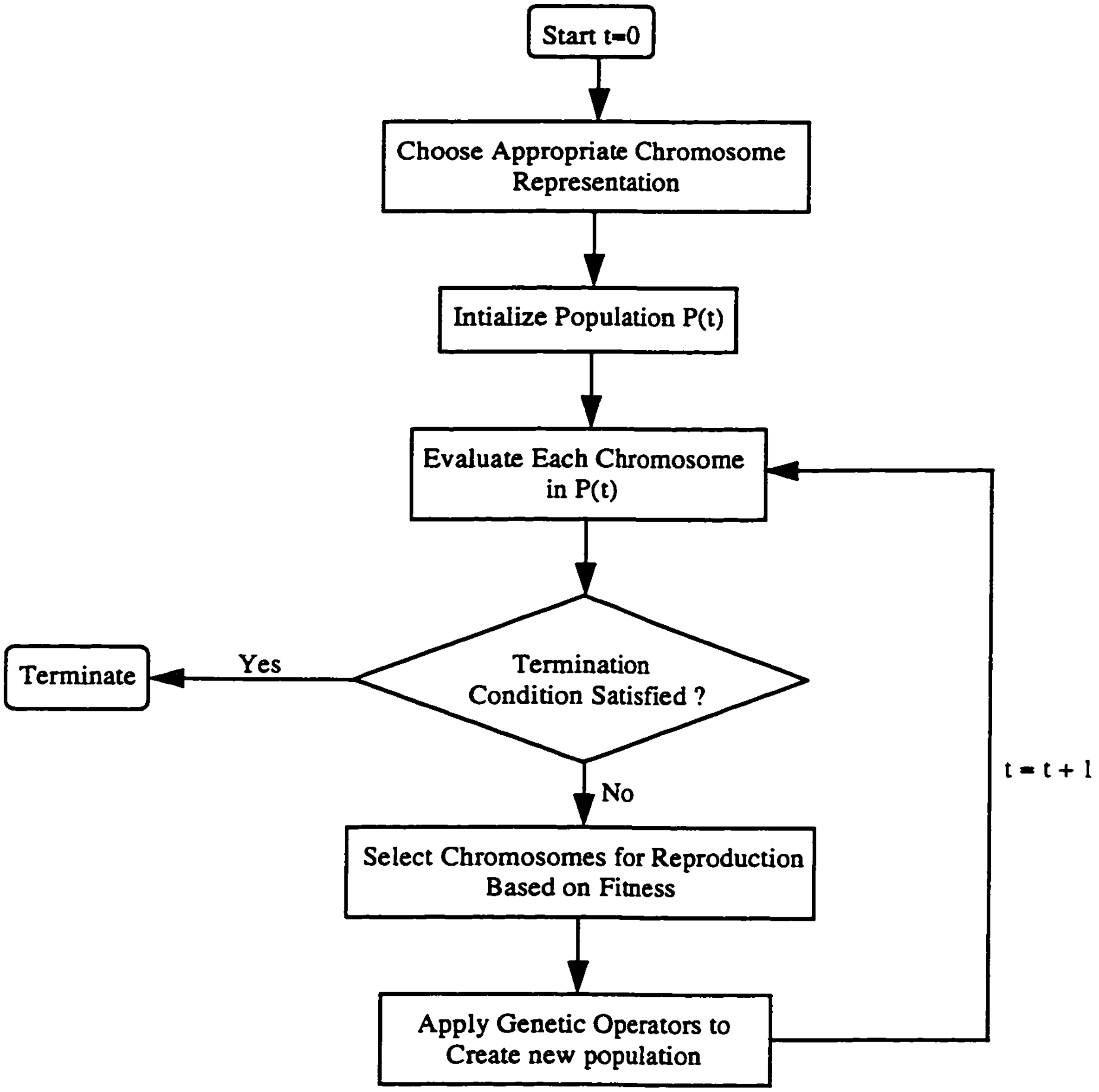


Figure 3.1 Structure of Standard Genetic Algorithms

(a) Reproduction is based on the ranking of individuals. All individuals are ranked according to their fitness values.

(b) One-at-a-time recombination and replacement occurs in one generation.

(c) Parents and Children both live and the worst solution in the population dies.

(d) Best found solutions are maintained in the population and the worst solution is eliminated and replaced.

(e) Parents are selected according to a selection bias. Since there is only one interaction at a time, it is very important to select the individuals from the population that will undergo recombination. Instead of crossover probability, a different parameter known as selection bias, is used in these algorithms.

(f) Duplicate strings are never created.

New evidence and arguments presented by Whitley (1989) suggest that allocating reproductive trials according to rank is superior to fitness proportionate reproduction.

The basic structure of steady-state genetic algorithm is shown in figure 3.2.

3.3 Genetic Algorithm Model

For Scheduling problems, genetic algorithms must have the following components:

(a) a genetic representation for potential solutions to the problem.

(b) a way to create an initial population of potential solutions

(c) an evaluation function for rating solutions in terms of their fitness.

(d) genetic operators that alter the composition of children during reproduction.

(e) values of various parameters that the genetic algorithms uses (population size, number of generation, crossover probabilities, etc.)

The following section provides a general discussion about these components

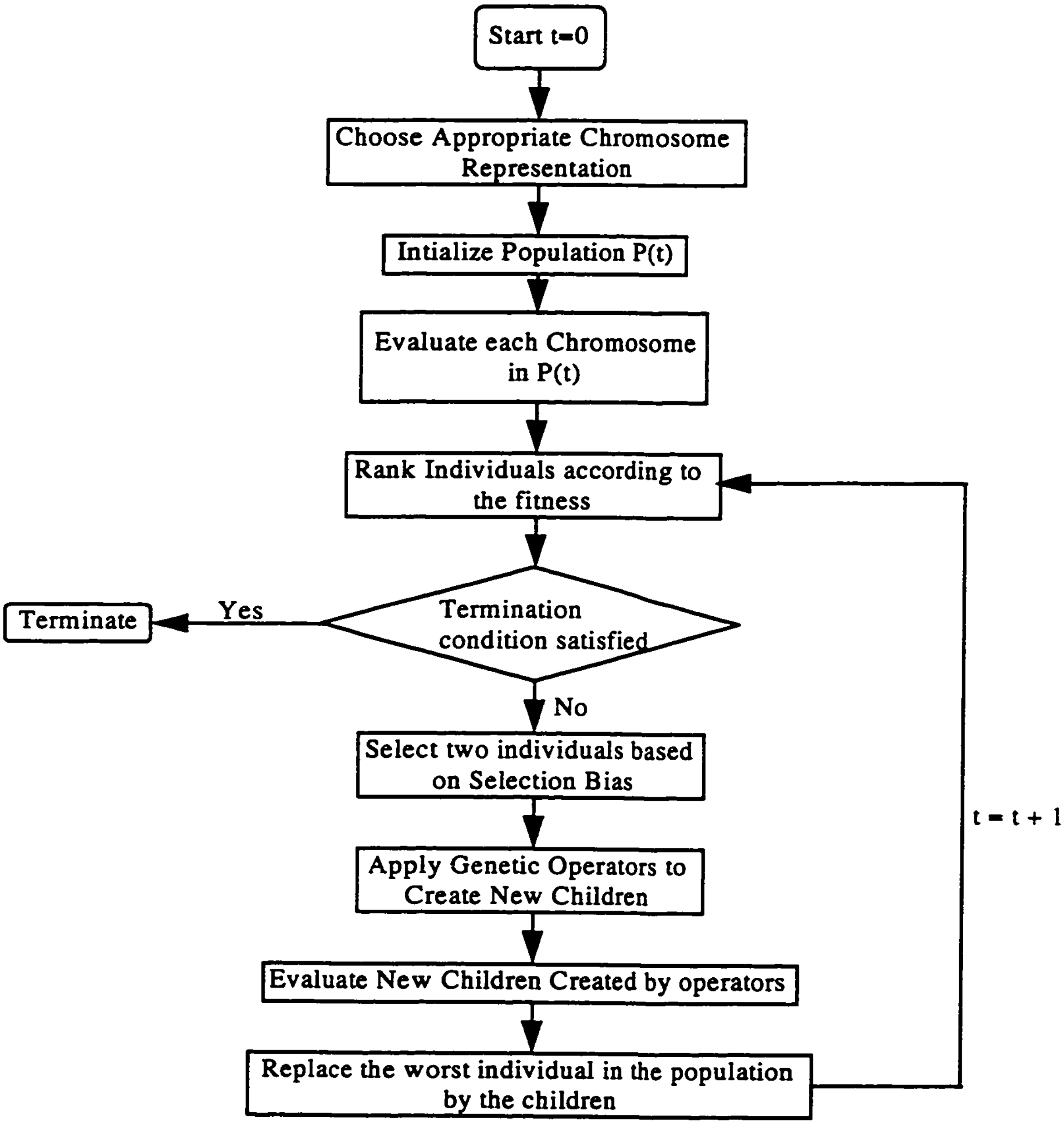


Figure 3.2 Structure of Steady State Genetic Algorithms

3.3.1 Chromosome Representation

Choosing an appropriate representation is the first step in applying genetic algorithms to an optimization problem. A number of representations such as adjacency representation (Grefenstette, et.al., 1985), permutation representation (Goldberg and Lingle, 1985), and ordinal representation (Grefenstette et.al., 1985) have been explored for such combinatorial-type scheduling problems. The genetic algorithms for DRC problems use a permutation type representation, here the sequence of operations is assumed to be fixed. In the case of routing flexibility each operation has a choice of one or more machines, and also because of workers capability to operate more than one machine; each operation has a choice of one or more machines and workers. The schedule will vary according to the operation-machine-worker assignments. The GA model employs a list of machines and workers as a chromosome (i.e. a schedule). Different combinations of operation-machine-worker assignments represent different production schedules and are used in the genetic population. This is shown in figure 3.3.

Operations	**O11**	**O12**	**O13**	**O21**	**O22**	**O23**
Machines	M2	M3	M2	M4	M2	M2
Worker	W2	W1	W2	W1	W3	W3

Figure 3.3 Chromosome Representation

The length of Chromosome is set equal to the total number of operations.

3.3.2 Initialization

The next important step is population initialization i.e. the creation of an initial resource assigned population. Population initialization can be done in two ways. It can be either generated randomly or a well adapted (seeded) population can be used as an initial population. For the DRC problem, an initial population of desired size has been generated randomly using a recursive procedure which lists all possible permutations randomly. An example of initial population thus generated by this procedure is shown in figure 3.4

Chromosomes	Population				
Schedule 1	M1 W2	M4 W1	M3 W2	M2 W2	M1 W1
Schedule 2	M2 W2	M3 W1	M2 W3	M4 W1	M2 W3
⋮			⋮		
Schedule N	M4 W3	M3 W2	M1 W2	M3 W1	M2 W3

Figure 3.4 List of Assigned Resources

The number of random schedules (N) to be generated in the population is specified by the user.

3.3.3 Evaluation Function

In order to find out the optimum solution of a problem, a GA starts from a set of assumed solutions (chromosomes) and evolves different but better sets of solutions over a sequence of iterations. In each generation (iteration) the objective function (fitness measuring criterion) determines the suitability of each solution and based on these values, some of them (which are called parent chromosomes) are selected for reproduction. The number of copies reproduced by an individual parent is expected to be directly proportional to its fitness value, thereby embodying the natural selection procedure, to some extent. The procedure thus selects the better (highly fitted) chromosomes and the worse ones are eliminated. Hence, the performance of a GA depends on the fitness evaluation criterion, to a large extent.

For the scheduling problem, the value of the evaluation function for a particular schedule is calculated by assigning operations to machines and workers. The evaluation function treats operations in an order consistent with the precedence relations of the problem. Once all the predecessors of an operation have been scheduled, the operation is said to be schedulable. This procedure is repeated until all operations have been scheduled. The genetic algorithm proceeds from generation to generation, saving the best schedules in each generation and getting rid of those schedules with objective function value lower than those of the saved ones.

3.3.4 Genetic Operators

In order to ensure that the problem space is comprehensively searched a means must be introduced in the genetic algorithms. This population variation is developed by using genetic operators. The classic genetic operators are described below.

3.3.4.1 Reproduction

The reproduction process copies individual strings (called parent chromosomes) into a tentative new population (known as mating pool) for generic operations. There are many different ways to implement reproduction, but almost any method which reproduces an individual is expected to be directly proportional to its fitness value, thereby mimicking the natural selection procedure to some extent. Roulette wheel parent selection are the most frequently used selection procedures.

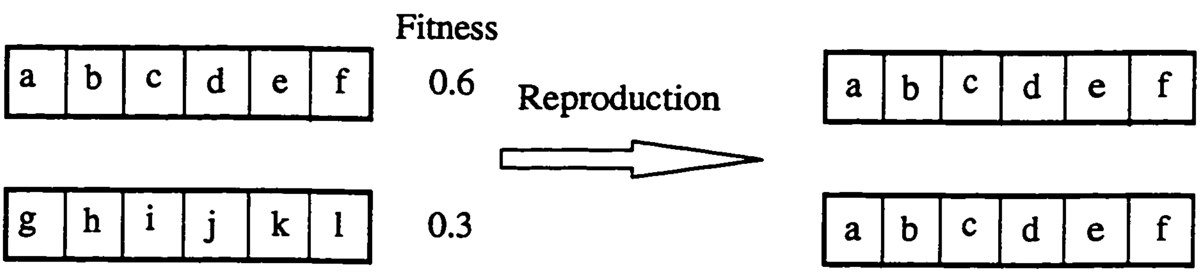


Figure 3.5 Reproduction Operator

3.3.4.2 Crossover

The main purpose of crossover is to exchange information between randomly selected parent chromosomes (schedules) with the aim of not losing any important information (i.e. minimum disruption of the structures of the chromosomes that are selected for a genetic operation). Actually, it recombines genetic material of two parent chromosomes to produce offsprings for the next generation. The crossover operation can be viewed as a three-step process. In the first step, pairs of chromosomes (called mating parents) are chosen from the mating pool. The second step determines the crossover point. The interchange of chromosome segments between mating pairs is done in the third step.

A great deal of research in the genetic algorithms field has been developed to investigate the effects of variations on the crossover operator, as well as to develop new kinds of crossover operators tailored to different encoding schemes and specific problem types. Scheduling problems usually employ list representations where the chromosomes are represented by the value of the gene itself (for example operation, machine, worker, etc.) making a chromosome representing a sequence of operations, machines, workers. Simple crossover operators for such problem would produce illegal schedules and for this reason, a variety of crossover operators have been developed for such ordering problems. Some of the commonly used operators are cyclic crossover, order crossover, partially matched crossover, edge recombination operator and reduce surrogate crossover. In this research we have used reduced surrogate crossover operator.

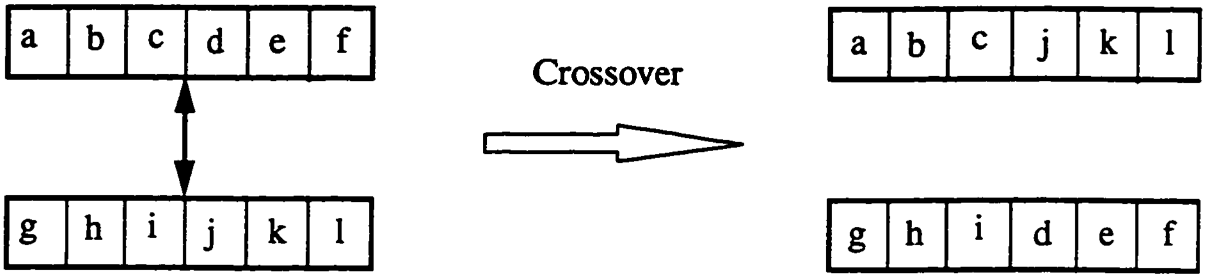


Figure 3.6 Crossover Operator

3.3.4.2.1 Reduced Surrogate Crossover

Crossover is usually implemented by choosing one crossover point at random, then exchanging segments between the two parent strings, thus forming new children which contain information from each of the two parent chromosomes. The problem of using such a crossover is that it can easily create duplicate chromosomes as indicated in figure 3.7. With reduced surrogate operator, the offsprings are guaranteed not to be duplicate of the parent chromosomes. First it identifies all position where the parent strings differ and then crossover is only allowed to occur in these positions.

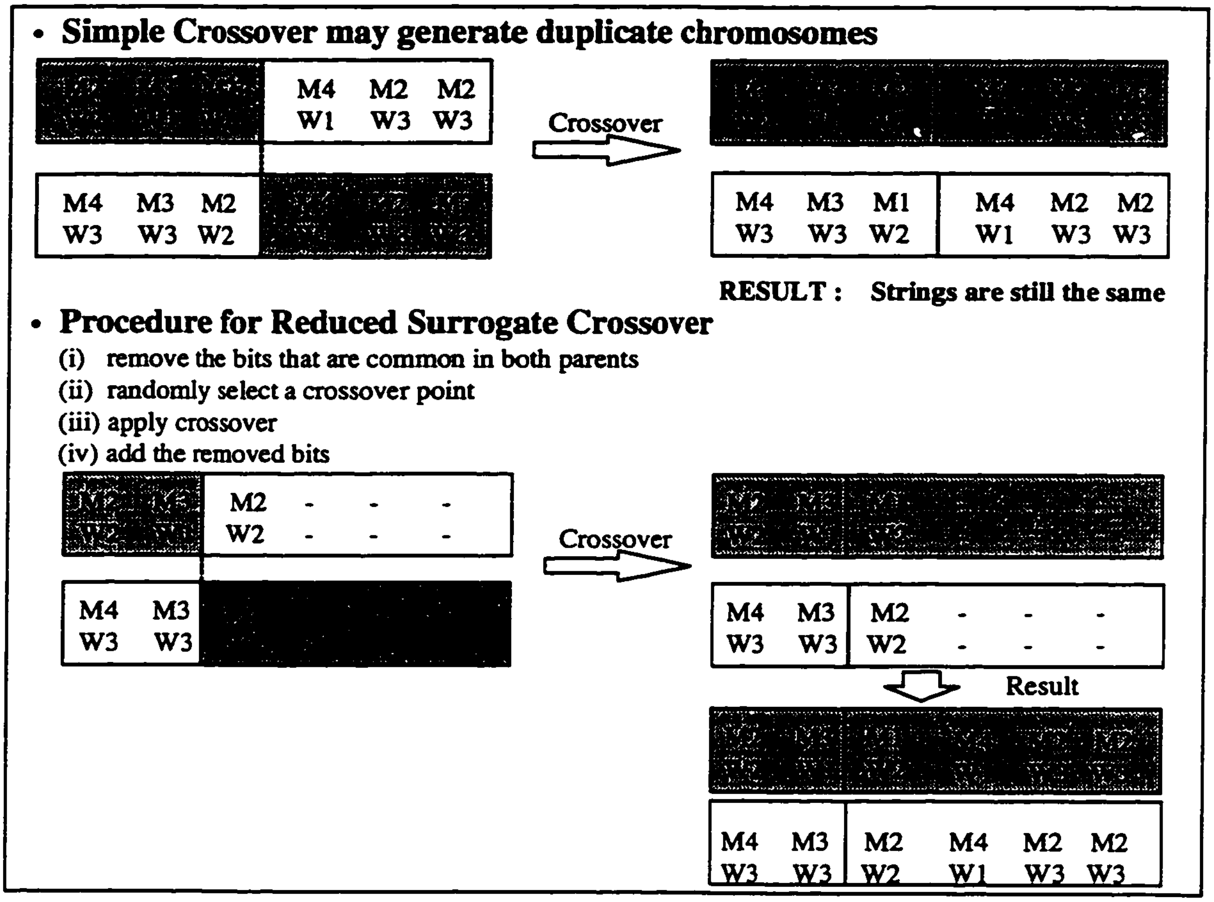


Figure 3.7 Reduced Surrogate Crossover

3.3.4.3 Mutation

The main aim of mutation is to introduce genetic diversity into the population. Sometimes it helps to regain the information lost in earlier generations. Like natural genetic system, mutation in genetic algorithms is also made occasionally. A random position of the string is selected and is replaced by another character. Normally mutation rate is kept fixed. To sustain diversity (which may be lost due to crossover and very low mutation rate) into the population, Whitley et.al (1990) proposed a technique called adaptive mutation, where the probability to perform the mutation operation is made to increase (instead of keeping it fixed) with the increase of genetic homogeneity in the population.

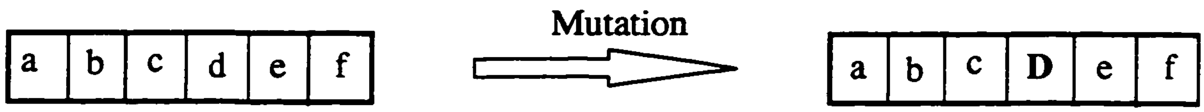


Figure 3.8 Mutation Operator

3.3.5 Parameter Values

The task of optimizing a complex problem like scheduling presents at least two levels of problems. First, a class of optimization algorithms must be chosen that is suitable for application to the system. Second, the various parameters that play a crucial role in the successful implementation of genetic algorithm are: crossover rate, mutation rate, population size and number of generations. The setting of these parameter values significantly affects the performance of GA. The problem of setting the parameters values has been studied extensively for bit string representation (Grefenstette, 1986). Very little work has been reported for other types of representation. In this research, an extensive set of computational experiments has been performed to determine the best value for the parameters.

CHAPTER IV

PROPOSED SCHEDULING METHODOLOGY

4.1 An Overview

How a company schedules its production activities can have a significant effect on its ability to meet its objectives including on-time delivery and resource utilization. Hence, our specific goal is to increase the throughput of the manufacturing process and effective utilization of resources by developing techniques for effectively scheduling the factory/shop floor.

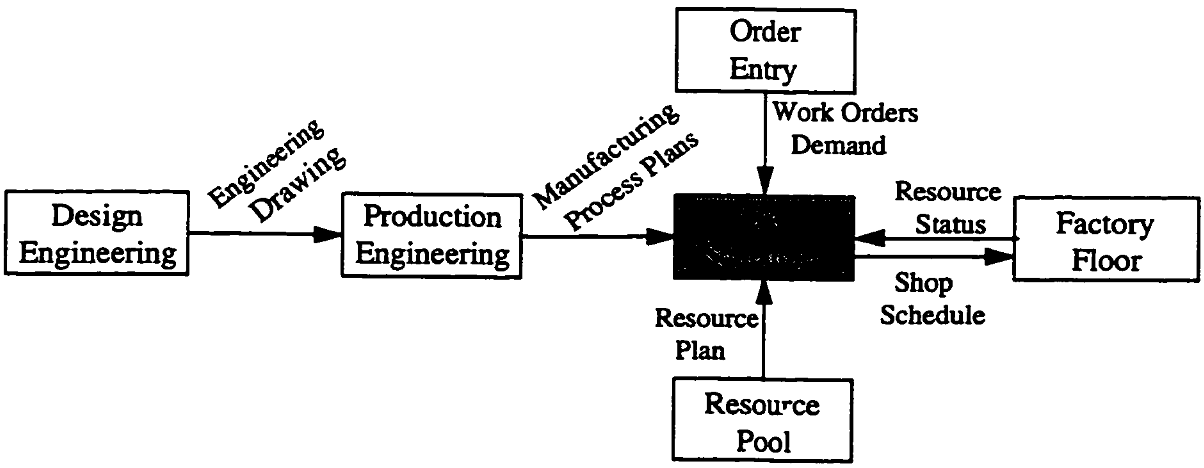


Figure 4.1 GA Process as Shop Scheduler

The genetic algorithm (GA) process, represents a potentially powerful tool for achieving improvements in total manufacturing time by reducing the time required following the engineering design phase. A simplified view of where GA fits is as shown in figure 4.1.

This chapter describes, a generic model for the scheduling problem, in which the shop capacity is constrained both by machines and workers. The genetic algorithm is best applied to scheduling problems when there are alternate routing and shop consisting of multi-skilled work force for each part. The jobs with alternate routing have a choice of alternate machines, which results in routing flexibility and has the potential of improving the shop performance by eliminating bottlenecks which are usually present when there are no alternate routings. Because of workers flexibility, which results in machine having alternate choice of workers, the capability to react quickly to temporarily unbalanced work loads exists.

	Part 1			Part 2		
Operations	**O11**	**O12**	**O13**	**O21**	**O22**	**O23**
Machines	M2 M3 M4	M1 M3 M4	M2 M1	M4 M3	M1 M2 M3	M2 M4

Figure 4.2 Task-Machine Information

The task of schedule optimization becomes difficult when there exists alternate routings for given jobs and when the shop has workers who are capable of operating more than one machine. The alternate routing gives rise to routing flexibility and multi-skilled workforce gives rise to worker flexibility, because of which the possible number of feasible assignment of resources grows exponentially and finding a satisfactory assignment of resources becomes a formidable task. This is shown in figure 4.2 which shows two parts each having three operations. Operation1 of Part1 can be performed on Machine M2, M3, M4; Operation2 of Part1 can be performed on Machine M1, M3, M4, and so on. Moreover from figure 4.3, Worker 1 and Worker 2 can work on Machine M1, Worker 2 and Worker 3 can work on Machine M2 and so on.

		Worker		
		W1	W2	W3
Machines	M1	1	1	0
	M2	0	1	1
	M3	1	0	1
	M4	1	0	1

Figure 4.3 Machine - Worker Information

Here it is assumed that the operations remain fixed while the sequence of machines on which these operations are performed and workers assigned varies. Hence, some of the possible assignment of resources in this case could be as shown in figure 4.4. The total number of possible assignment of resources in this case would be 13824 (3x2x3x2x2x2x2x2x3x2x2x2). If it was assumed that worker is not the constraining factor and each worker can work on only one machine, the total number of possible assignment of resources for the same case will be 216 (3x3x2x2x3x2). Therefore, by just adding one complexity to the problem the search space increased from 216 to 13824.

	O11	O12	O13	O21	O22	O23
1	M2 L2	M3 L1	M2 L2	M4 L1	M2 L3	M2 L3
2	M2 L2	M3 L1	M2 L3	M4 L3	M2 L2	M2 L3
3	M4 L3	M3 L3	M1 L2	M3 L1	M3 L1	M4 L3

Figure 4.4 List of Assigned Resources

4.2 Single v/s Dual resource constrained shop

In both Single resource constrained (Machine-limited) and Dual resource constrained (Machine & Worker limited) systems, the execution phase of scheduling includes priority sequencing and resource allocation. For a single resource constrained system the constraining resource is machines. The assignment of this resource to tasks is process plan dependent. The process plan determines the machines required for all task in the system. The start and completion time of each task is determined by the dispatching rule and the processing time. For GA formulation the assignment of machines to tasks can be done using Task-Machine Information matrix. If a task can be performed on a machine then the value in that matrix cell is 1, otherwise zero as shown in figure 4.2. The sub-chromosome representing assignment of machines to tasks is represented as in figure 4.5. In machine-limited system , dispatching rules are sufficient for executing a schedule.

Operations	**O11**	**O12**	**O13**	**O21**	**O22**	**O23**
Machines	M2	M3	M2	M4	M2	M2

Figure 4.5 Sub-chromosome representing Machine allocation

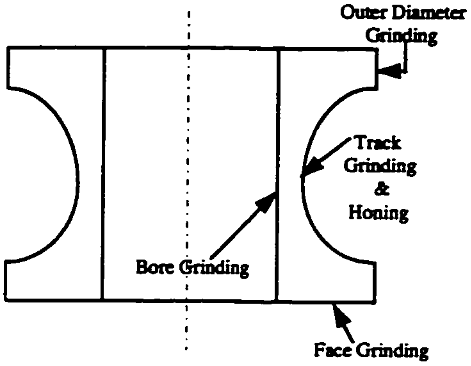
For a dual resource constrained system, a combination of machine assignment, job dispatching rules (for prioritizing jobs at machines) and worker assignment to machines are required. In this type of system, allocation of machine resource is done first as mentioned above. The allocation of workers is dependent on the machine and worker's skills (worker flexibility). The worker flexibility determines what machine a worker is able to operate. This information is provided by the Worker-Machine Information matrix as shown in figure 4.3. For this type of system the start and completion time of the task is determined by dispatching rules, processing time and the efficiency of the worker operating the machine. The start time of the task is also dependent on the availability of assigned worker. For GA formulation first the machine is assigned to a task using the Task-Machine Information matrix, then for the assigned machine to task, worker assignment is done using Worker-Machine Information matrix. The chromosome representing the Task-Machine-Worker is as shown in figure 4.6. The level of complexity of shop floor control in the execution phase is greater in dual resource constrained systems compared with single resource constrained systems.

Operations	O11	O12	O13	O21	O22	O23
Machines	M2	M3	M2	M4	M2	M2
Worker	W2	W1	W2	W1	W3	W3

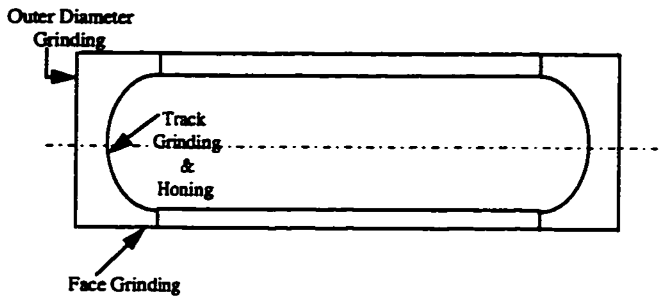
Figure 4.6 Chromosome representing Machine-Worker allocation

4.3 Problem Description

The DRC Scheduling problem may be stated as : "Given process plans for each part and shop consisting of flexible workforce, the objective is to find a feasible schedule for given set of job orders such that some given performance criterion is optimized." The prerequisites of this problem is the process plan for each job, the number and type of jobs, number of tasks in each job, number and types of machines available, number and types of workers available, processing time and setup time of tasks on machines, workers efficiency information, order due dates, release time of jobs onto the shop floor and the performance measure to be optimized.



(a) Inner Race



(b) Outer Race

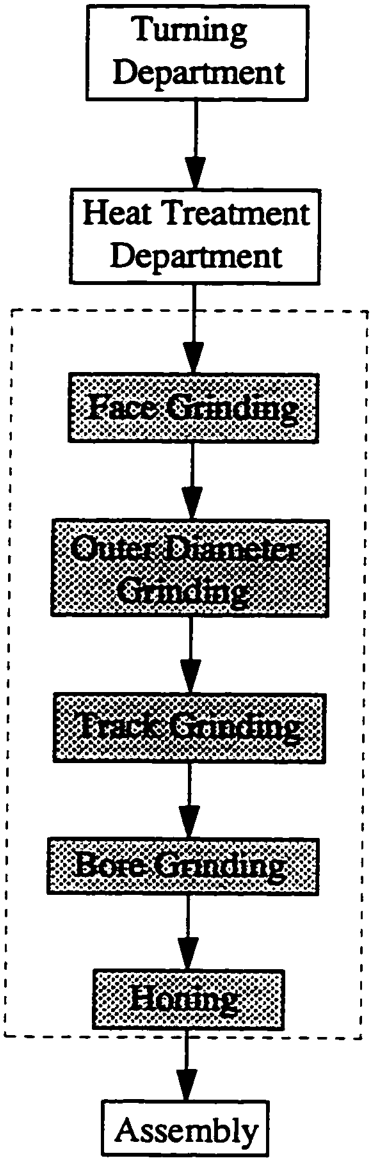
Figure 4.7 Bearings and Bearings Parts

This research considers a hypothetical job-shop involved in machining of the inner and outer race of ball bearings (figure 4.7). The shop is comprised of three different departments before the job reaches the assembly department as shown in figure 4.8.

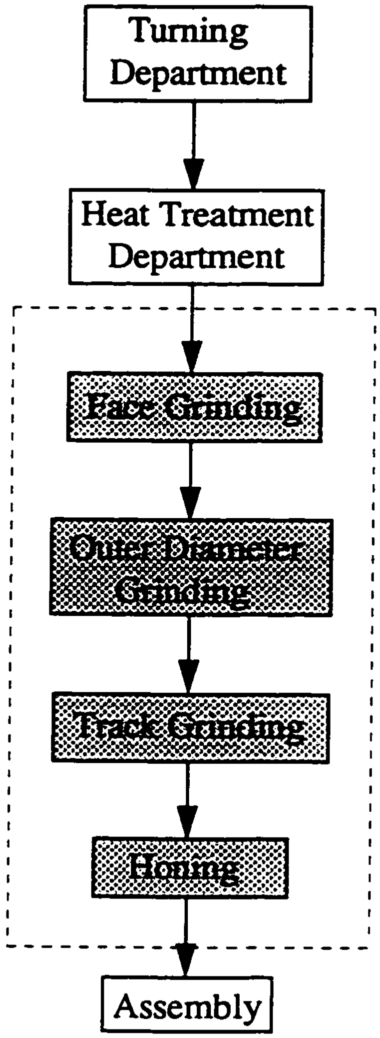
Figure 4.8 Job Shop Model for Machining of Bearing Inner Race and Outer Race

This research focuses on the scheduling of jobs in Grinding Department. The grinding department consists of 10 machines, and each machine is assumed to be unique, and has different processing capabilities. The job routings is hybrid which lies between the extremes of a pure job shop and a pure flow shop. The process plans for inner race and outer race is illustrated in figure 4.9.

The model is tested for various workers staffing levels, of 50%, 60%, 70%, 80% each having flexibility level of 2 and 100% with flexibility level of 1. A staffing level of 90% is not considered since the benefits derived from increasing staffing from 80% to 90% is minimal (Felan III et.al. 1993). A staffing level of 50% means that five workers are assigned to ten machines, while flexibility of two means that each worker is capable of operating two machines. This staffing level and worker flexibility is similar to that used in previous studies (Fryer 1974, Gunther 1979, Treleven & Elvers 1985, Felan III et.al. 1993). Due dates are assigned based on the total work content (TWK) method. The allowance factor (due date tightness) required by TWK is set equal to one, which represents a set of tight due dates, to provide for a more stark set of comparisons.



(a) Process Plan for Inner Race



(b) Process Plan for Outer Race

Figure 4.9 Process Plans for Inner Race and Outer Race

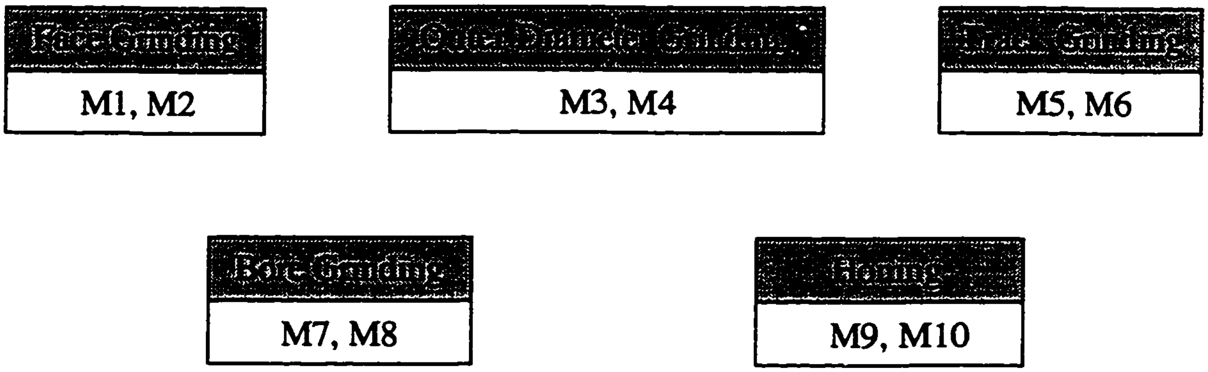


Figure 4.10 Hypothetical Job Shop for Grinding Department

In this research six different dispatching rules are employed for various staffing levels (70% and 100%) and performance of each dispatching rule is judged against the performance criteria.

4.3.1 Shop Performance Measures

In studies such as this, it is difficult to use specific cost-oriented criteria to measure shop performance due to varying cost-structures for different organizations. Two criteria are used in this research: first criterion used is work-in-process and due-date performance which represents the primary drivers for measuring manufacturing performance, and second cost related criterion is worker utilization and machine utilization which to many organizations, may be the single most important indicator of cost. Managers have tried to avoid idle workers and machines in order to maximize worker utilization and machine utilization. The most common indicator to measure shop performance using work-in-process (WIP) as the criterion is mean flow time (F). Conway et.al. (1987) described three principal types of cost relevant to sequencing decisions: cost of inventory, lateness and utilization. It was shown that reducing mean flow time is tantamount to reducing average work-in-process inventory, reducing mean lateness and permitting a facility to be efficiently utilized. Typical measures considered in this research are as follows:

(1) Minimize Mean Flow time and Makespan

$$\text{Mean Flow Time} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=J_i} C_{ij} \tag{3.1}$$

$$\text{Makespan} = \max\,(C_{ij}) \qquad \forall\, i \quad \text{and} \quad j = J_i \tag{3.2}$$

(2) Minimize Mean and Maximum Waiting time

$$\text{Waiting Time} = W_{ij} = \sum_{j=J_i} C_{ij} - \sum_{j=0} r_{ij} - \sum_{j=0}^{J_i} t_{ij} \qquad \forall\, i \tag{3.3}$$

$$\text{Mean Waiting Time} = \frac{1}{n}\sum_{i=1}^{n} W_{ij} \tag{3.4}$$

$$\text{Maximum Waiting time} = \max\,(W_{ij}) \qquad \forall\, j = J_i \text{ and } i \tag{3.5}$$

(3) Minimize Mean and Maximum Tardiness

Tardiness is the positive difference between the jobs completion time and its due date.

$$\text{Tardiness} = T_{ij} = \sum_{j=J_i} C_{ij} - d_i \qquad \forall\, i \tag{3.6}$$

$$\text{Mean Tardiness} = \frac{1}{n}\sum_{i=1}^{n} T_{ij} \tag{3.7}$$

$$\text{Maximum Tardiness} = \max\,(T_{ij}) \qquad \forall\, j = J_i \text{ and } i \tag{3.8}$$

(4) Maximize Utilization of available resources

The percent utilization of individual machines and workers is calculated based on the maximum flow time, i.e. for a schedule, the individual and average resource utilization is calculated as follows:

$$\text{Utilization (Machine)} = U_k = \frac{\text{Total machine busy time}}{\max(C_{ij})} \quad \forall k \qquad (3.9)$$

$$\text{Average Machine Utilization} = \frac{1}{m}\sum_{k=1}^{m} U_k \qquad (3.10)$$

The worker busy time is equal to machine busy time, because it is assumed that worker is occupied during loading, processing time and unloading time.

$$\text{Utilization (Worker)} = U_l = \frac{\text{Total Worker busy time}}{\max(C_{ij})} \quad \forall l \qquad (3.11)$$

$$\text{Average Worker Utilization} = \frac{1}{p}\sum_{l=1}^{p} U_l \qquad (3.12)$$

All the criteria are evaluated in the evaluation function of the genetic algorithms. The evaluation function gives a complete assignment of resources and an associated performance criterion value.

4.3.2 Assumptions

Much of the scheduling work done so far is based on the assumption that the only scarce resource is equipment, and worker is not treated as the constraining factor, which is not practical in real life. Some of the assumptions considered for this research are as follows:

(a) A Task can be performed on one or more different (alternate) machines.

(b) Workers may be transferred from one machine to another, subject to certain restrictions (depending on skills and familiarity with machines).

(c) Workers are equally efficient at all machines, thus no loss in efficiency results.

(d) Pre-emption is not allowed (operations are completed without interruption).

(e) Machines breakdown is not considered.

(f) Each operation must be completed before a subsequent operation can begin.

(g) Processing times and Setup times are deterministic and known in advance.

(h) Transportation time between facilities is negligible or included in processing time.

(i) Worker transfer delay not considered.

(h) Worker is occupied with job/machine for whole of the loading time, processing time and unloading time.

(i) Batch splitting is not allowed for scheduling purposes.

4.4 Scheduling Algorithm

The evaluation function determines the quality of solutions in the genetic population. In each generation, the value of the evaluation function for a particular schedule is calculated by using the following scheduling algorithm (Nasr and Elsayed 1990) modified to handle dual resource constraints.

(1) Calculate ready time for all operations

Let $t = 0$, $a_k = 0$, and $b_l = 0$

Beginning with S as an empty set while S' includes all the operations.

Set $r_{i1} = 0$, for all i

$r_{ij} = r_{i(j-1)} + t_{i(j-1)k}/E_{lk}$, for all $i = 1, 2, \ldots, n$, and $j = 2, 3, \ldots, J_i$

(2) If there is any $[O_{ij} \in S' \mid r_{ij} \leq t]$ go to step (3).

Otherwise : Set $t = \min(r_{ij})$.

(3) Place all $[O_{ij} \in S' \mid r_{ij} \leq t]$ in a set S".

(4) For each operation $O_{ij} \in S"$, calculate a priority index according to the dispatching rule specified by the user. The operation with the smallest priority index is assigned to the respective machine and worker. Calculate the completion time of each operation

$C_{ij} = \max(\max(r_{ij}, a_k), b_l) + t_{ijk}/E_{lk}$

(5) For each operation in the set S" :

(a) Remove the operation O_{ij} from the set S' and the set S".

(b) Place O_{ij} in the set S.

(c) Determine :

$S_{ij} = \max \{ r_{ij}, a_k, b_l \}$

$C_{ij} = S_{ij} + t_{ijk}/E_{lk}$

$r_{i(j+h)} = r_{i(j+h)} + [C_{ij} - r_{i(j+1)}]$ for $h = 1, 2, \ldots, (J_i - j)$

(d) Set $a_k = C_{ij}$ and $b_l = C_{ij}$

(6) Check if S' is empty, if not then go to step (2) and repeat the procedure until the schedule is complete.

The genetic algorithm proceeds from generation to generation saving the best schedules in each generation and getting rid of those schedules whose objective function values are low compared to the saved ones. In each generation the population size remains fixed.

4.5 Genetic Algorithm Model

The genetic algorithm model is developed incorporating the design issues as discussed below:

(1) ***Chromosome Representation*** : The model employs a list of machines and workers as a chromosome i.e. a schedule. The length of chromosome is equal to the total number of operations. Different combinations of operation-machine-worker assignments results in different schedules.

(2) ***Initialization*** : Select the initial parameters and create an initial diversified population of schedules.

(a) Set the values for population size (PSIZE), selection bias (SBIAS), adaptive mutation (AMUT), and number of generations (NGEN).

Selection Bias is a floating point number which specifies the amount of preference to be given to superior individuals in a genetic population.

(b) Read the processing times, Setup times, due dates and ready times for all the jobs.

(c) Read the worker efficiency for all the machines in the system.

(d) Create an initial population of schedules of size PSIZE and call it 'oldpop'.

(e) Calculate the objective function values for all the schedules in the population using the scheduling algorithm.

(f) Set NGEN = 1 (i.e. current generation = 1).

(3) ***Recombination*** : Apply recombination operator to the 'oldpop' to form a new population.

(a) Two parents are selected from the oldpop based on the specified selection bias. They are called 'mom' and 'dad'. Parents are selected based on the following algorithm (Whitley, 1989):

$$\text{index} = \frac{\text{PSIZE} \times (\text{SBIAS} - \text{sqrt}(\text{BIAS}^2 - 4.0 \times (\text{SBIAS} - 1) \times \text{random}(\,)))}{2.0 \,/\, (\text{BIAS} - 1)}$$

where index is the schedule number to be selected from the sorted population.

(b) The reduced surrogate operator is applied to the two selected parents to form a new child.

(c) Calculate the objective function value for this child.

(d) If the objective function value of the child is better than any of the schedules in the population, then insert the child in the population at the appropriate place and remove the worst schedule from the population.

(4) ***New Generation*** : Evaluate the current generation number to determine the next step.

(a) If GEN < NGEN, then GEN is incremented by one and the current population becomes oldpop. GO TO Step(3).

(b) If GEN = NGEN, then STOP.

The best schedule would be the schedule in the current population with the highest objective function value.

4.6 Genetic Algorithm Parameter Setting

The selection of best combination of genetic algorithm parameters is the most difficult and time consuming. The parameters used for this problem are population size (PSIZE), selection bias (SBIAS) and the adaptive mutation (AMUT) rate. To find the best parameters values, extensive experiments were conducted with different combinations of parameter values. The parameter values is given in figure 4.11.

The population size was varied from 40 to 70 with an interval of 10. The selection bias (SBIAS) was varied from 1.1 to 1.9 in a step of 0.1 and the mutation rate was varied from 0.1 to 0.9 in a step of 0.2. The number of generations was varied between 200 to 500. The example (A1) used for experimentation consists of four jobs, inner race and outer race of one bearing type; and inner race, outer race for a second bearing type. The process plan of inner race for both types of bearings are the same, but the processing time on each machine is different for both the bearing types. Similar is the case for both of the outer race. Each task has a choice of alternate machines on which processing takes place and has a choice of alternate worker who can perform the task on that machine. The characteristics of this problem such as processing time, setup time, task-machine information, worker-machine information are given in Appendix A.

Parameters	Levels
Population Size (PSIZE)	40 50 60 70
Selection Bias (SBIAS)	1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
Adaptive Mutation (AMUT)	0.1 0.3 0.5 0.7 0.9

Figure 4.11 Experimental design parameters for GAs

	Task 1	Task 2	Task 3	Task 4	Task 5	Quantity
Job Type1	Face Grinding	Outer Dia. Grinding	Track Grinding	Bore Grinding	Honing	50
Job Type2	Face Grinding	Outer Dia. Grinding	Track Grinding	Honing	-	50
Job Type3	Face Grinding	Outer Dia. Grinding	Track Grinding	Bore Grinding	Honing	50
Job Type4	Face Grinding	Outer Dia. Grinding	Track Grinding	Honing	-	50

Table 4.1 Task Precedence and Order quantity

The example (A1) is run for different combinations of population size, selection bias and the mutation rate. The performance measure and the dispatching rule used to minimize are the mean flow time and the shortest processing time. Various worker staffing levels were considered from 50% staffing level to 100% (inflexible work force) staffing level.

		Population Size (PSIZE)							
		PSZIE = 40		PSIZE = 50		PSZIE = 60		PSIZE = 70	
SBIAS	AMUT	Value	GEN	Value	GEN	Value	GEN	Value	GEN
1.1	0.1	2525.0	200	2587.5	200	2562.5	200	2562.5	200
	0.3	2575.0	300	2550.0	300	**2487.5**	300	2500.0	300
	0.5	2512.5	500	2550.0	500	**2487.5**	500	**2487.5**	500
	0.7	**2487.5**	500	**2487.5**	500	2500.0	500	2512.5	500
	0.9	2525.0	500	**2487.5**	500	2537.5	500	**2487.5**	500
1.2	0.1	2512.5	200	2525.5	200	2575.5	200	2512.5	200
	0.3	**2487.5**	300	**2487.5**	300	2512.5	300	2512.5	300
	0.5	**2487.5**	500	2500.0	500	**2487.5**	500	**2487.5**	500
	0.7	2500.0	500	2550.0	500	**2487.5**	500	**2487.5**	500
	0.9	**2487.5**	500	2512.5	500	**2487.5**	500	**2487.5**	500
1.3	0.1	2537.5	200	2562.5	200	2550.0	200	2525.0	200
	0.3	2537.5	300	2537.5	300	2550.0	300	2537.5	300
	0.5	**2487.5**	500	2562.5	500	**2487.5**	500	**2487.5**	500
	0.7	2537.5	500	**2487.5**	500	2512.5	500	**2487.5**	500
	0.9	**2487.5**	500	2500.0	500	2500.0	500	2500.0	500
1.4	0.1	**2487.5**	200	2575.0	200	2575.0	200	2537.5	200
	0.3	**2487.5**	300	2525.0	300	2550.0	300	2525.0	300
	0.5	2500.0	500	2500.0	500	2512.5	500	**2487.5**	500
	0.7	**2487.5**	500	2512.5	500	2562.5	500	**2487.5**	500
	0.9	**2487.5**	500	2500.0	500	2500.0	500	**2487.5**	500
1.5	0.1	2525.0	200	2550.0	200	2600.0	200	2562.5	200
	0.3	2537.5	300	2537.5	300	2575.0	300	2537.5	300
	0.5	2537.5	500	2500.0	500	2525.0	500	2525.0	500
	0.7	2512.5	500	2575.0	500	2500.0	500	**2487.5**	500
	0.9	2500.0	500	2537.5	500	**2487.5**	500	2550.0	500
1.6	0.1	2512.5	200	2562.5	200	2575.0	200	2525.0	200
	0.3	2525.0	300	2525.0	300	2537.5	300	**2487.5**	300
	0.5	2562.5	500	**2487.5**	500	**2487.5**	500	**2487.5**	500
	0.7	**2487.5**	500	2500.0	500	2562.5	500	**2487.5**	500
	0.9	**2487.5**	500	2500.0	500	2525.0	500	2497.5	500

Best Value for Mean Flow Time is 2487.5 sec.

Table 4.2 Experimental Results for Mean Flow Time with Staffing Level 50%

		Population Size (PSIZE)							
		PSZIE = 40		PSIZE = 50		PSZIE = 60		PSIZE = 70	
SBIAS	AMUT	Value	GEN	Value	GEN	Value	GEN	Value	GEN
1.7	0.1	2550.0	200	2537.5	200	2512.5	200	2525.0	200
	0.3	2512.5	300	2525.0	300	2512.5	300	2500.0	300
	0.5	2512.5	500	2500.0	500	**2487.5**	500	2525.0	500
	0.7	2537.5	500	2500.0	500	**2487.5**	500	2525.0	500
	0.9	2525.0	500	**2487.5**	500	**2487.5**	500	2562.5	500
1.8	0.1	2525.0	200	2575.0	200	2500.0	200	2550.0	200
	0.3	2500.0	300	**2487.5**	300	**2487.5**	300	2537.5	300
	0.5	**2487.5**	500	2512.5	500	**2487.5**	500	**2487.5**	500
	0.7	2500.0	500	2525.0	500	**2487.5**	500	2537.5	500
	0.9	2500.0	500	2512.5	500	2575.0	500	2500.0	500
1.9	0.1	2512.5	200	2562.5	200	2537.5	200	2525.0	200
	0.3	2550.0	300	2550.0	300	**2487.5**	300	**2487.5**	300
	0.5	2562.5	500	2537.5	500	2512.5	500	**2487.5**	500
	0.7	2537.5	500	2537.5	500	2500.0	500	**2487.5**	500
	0.9	2500.0	500	2500.0	500	**2487.5**	500	**2487.5**	500

Best Value for Mean Flow Time is 2487.5 sec

Table 4.2 Experimental Results for Mean Flow Time with Staffing Level 50%

The characteristics of workers for various staffing levels for the problem considered are given in Appendix A. The search process begins with the proper representation of schedule, generation of schedule population and evaluation of each schedule in the population, and the application of genetic operators to this population for improving schedules. The process continues for a specified number of generations. The results for worker staffing equal to 50% with flexibility level 2 and worker staffing level equal to 70% with flexibility level 2 are given in Table 4.2 and Table 4.3.

4.7 Computational Experience

For the problem considered, it was observed that the optimal value of mean flow time is 2487.5 time units for worker staffing level of 50%, which was obtained for several cases with different combination of PSIZE, SBIAS, and AMUT. The solution converged to optimal value for a population size of 60 and SBIAS of 1.3 and AMUT of 0.1 and it took 400 generations to converge. The best combination that took the least number of generations which was 200, is a SBIAS of 1.4 and AMUT of 0.1. Other values of population size, SBIAS and AMUT of course attained the optimal value, but it took more number of generations to converge. As far as selection bias is concerned, all values of SBIAS eventually yielded good results, some for a smaller population size and some for large population size. The best population size observed was 60, larger population size of value greater than 70 was tried, but no improvements was observed over this optimal value. Larger population size resulted in large number of generations to converge to optimal value, resulting in large computational time. As far as mutation rate is concerned, a low value of 0.3 or 0.1 was found to be a better choice.

For worker staffing level of 70% the optimal value found was 2287.5 units. For this, the best combination of parameter values found to converge to an optimal value were PSIZE of 60, SBIAS of 1.5 and AMUT of 0.3, and number of generations it took to converge was 300 generations. It can be observed from the previous discussion that for the same problem by only varying the worker characteristics, the optimal value was found with different combinations of parameter values. Hence, it can be concluded from this discussion that these parameters of PSIZE, SBIAS, and AMUT are very much dependent on the probiem chosen.

		Population Size (PSIZE)							
		PSZIE - 40		PSIZE - 50		PSZIE - 60		PSIZE - 70	
SBIAS	AMUT	Value	GEN	Value	GEN	Value	GEN	Value	GEN
1.1	0.1	2450.0	200	2400.0	200	2375.0	200	2437.5	200
	0.3	2425.0	300	2337.5	300	2350.0	300	2337.5	300
	0.5	2325.0	500	2337.5	500	2375.0	500	**2287.5**	500
	0.7	**2287.5**	500	2337.5	500	2337.5	500	2337.5	500
	0.9	2337.5	500	2387.5	500	2375.0	500	2375.0	500
1.2	0.1	2437.5	200	2387.5	200	2400.0	200	2425.0	200
	0.3	2337.5	300	2350.0	300	2362.5	300	2350.0	300
	0.5	2387.5	500	2337.5	500	**2287.5**	500	**2287.5**	500
	0.7	2337.5	500	2375.5	500	2337.5	500	2375.0	500
	0.9	2412.5	500	2375.0	500	2375.0	500	**2287.5**	500
1.3	0.1	2337.5	200	2375.0	200	2312.5	200	2325.0	200
	0.3	2362.5	300	2337.5	300	2325.0	300	2387.5	300
	0.5	2350.0	500	2337.5	500	2337.5	500	2412.5	500
	0.7	2315.5	500	**2287.5**	500	2375.0	500	**2287.5**	500
	0.9	2425.0	500	2387.5	500	2375.0	500	2287.5	500
1.4	0.1	2400.0	200	2387.5	200	2312.5	200	2387.5	200
	0.3	2375.0	300	2337.5	300	2337.5	300	2325.0	300
	0.5	2362.5	500	2325.0	500	2362.5	500	**2287.5**	500
	0.7	2325.0	500	2350.0	500	2362.5	500	2325.0	500
	0.9	2400.0	500	2362.5	500	2350.0	500	2287.5	500
1.5	0.1	2387.5	200	2362.5	200	2325.0	200	2312.5	200
	0.3	2337.5	300	2337.5	300	**2287.5**	300	2337.5	300
	0.5	2362.5	500	2375.0	500	2337.5	500	2312.5	500
	0.7	2337.5	500	2350.0	500	2325.0	500	2300.0	500
	0.9	2387.5	500	2325.0	500	2350.0	500	2387.5	500
1.6	0.1	2400.0	200	2350.0	200	2375.0	200	2337.5	200
	0.3	2337.5	300	2387.5	300	2337.5	300	2312.5	300
	0.5	2412.5	500	2337.5	500	2350.0	500	2350.0	500
	0.7	2375.0	500	2350.0	500	2375.0	500	**2287.5**	500
	0.9	2350.0	500	2325.0	500	2375.0	500	2375.0	500

Best Value for Mean Flow Time is 2287.5 sec

Table 4.3 Experimentation Results for Mean Flow Time with Staffing Level 70%

		Population Size (PSIZE)							
		PSZIE = 40		PSIZE = 50		PSZIE = 60		PSIZE = 70	
SBIAS	AMUT	Value	GEN	Value	GEN	Value	GEN	Value	GEN
1.7	0.1	2337.5	200	2337.5	200	2337.5	200	2375.0	200
	0.3	2337.5	300	2387.5	300	2375.0	300	2312.5	300
	0.5	2337.5	500	2337.5	500	2337.5	500	2325.0	500
	0.7	2312.5	500	2362.5	500	2387.5	500	2362.5	500
	0.9	2337.5	500	2412.5	500	2325.0	500	2312.5	500
1.8	0.1	2375.0	400	2337.5	400	2375.0	400	2337.5	400
	0.3	2325.0	400	2337.5	400	2300.0	400	2350.0	400
	0.5	2387.5	400	2375.0	400	2287.5	400	2287.5	400
	0.7	2350.0	400	2387.5	400	2350.0	400	2375.0	400
	0.9	2400.0	400	2325.0	400	2337.5	400	2312.5	400
1.9	0.1	2387.5	400	2375.0	400	2337.5	400	2350.0	400
	0.3	2375.0	400	2387.5	400	2362.5	400	2350.0	400
	0.5	2375.0	400	2337.5	400	2362.5	400	2337.5	400
	0.7	2337.5	400	2312.5	400	2350.0	400	2337.5	400
	0.9	2337.5	400	2437.5	400	2375.0	400	2300.0	400

Best Value for Mean Flow Time is 2287.5 sec

Table 4.3 Experimentation Results for Mean Flow Time with Staffing Level 70%

Different parameter values of GA yield significant different results for the same problem, and hence several experiments should be run before a decision can be made on exact parameter values. Since computation times for genetic algorithms are usually small, it may not be difficult task to conduct experiments. Similar trends were obtained when different worker staffing level such as 60% staffing level, 80% staffing level and for machine constrained shop (with 100% worker staffing level and inflexible work force).

4.8 Results and Discussions

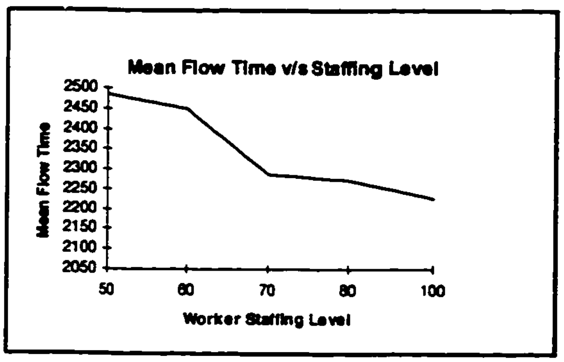
Table 4.4 presents the experimentation results for all performance measures with an objective to minimize mean flow time, for all worker staffing levels. The appropriate strategy for an existing factory to use in an effort to increase process flexibility is dependent on the reliance by the company managers using specific cost-oriented criteria to measure shop performance. For the hybrid job-shop modeled in this research it can be observed from table 4.4 that from a staffing level of 60% (flexibility level 2), a dramatic improvement in mean flow time and other performance measures can be achieved by increasing staffing level to 70% (flexibility level 2). However, increased costs in the form of direct worker cost and worker utilization results. When worker staffing level is 70% and worker flexibility is two, the mean flow time is 2287.5. This value is significantly different from the corresponding value of mean flow time of 2450 for worker staffing level of 60%. From table 4.5 it can be observed that there was improvement of 6.6% in mean flow time performance measure, resulting in an increase of direct worker cost of 16.66%. But for subsequent increase of worker staffing level from 70% to 80% there was not much improvement in the mean flow time and other performance measures. Increasing the direct worker cost by 14% due to increasing staffing from 70% to 80% resulted in marginal gain of 0.05% for mean flow time.

Staffing Level	Mean FT sec	Max FT sec	Mean T sec	Max T sec	Mean WT sec	Max WT sec	Mach Util. %	Worker Util. %
50%	2487.5	3600.0	325.0	900.0	500.0	1150.0	22.1	44.2
60%	2450.0	3500.0	262.5	800.0	462.5	1150.0	22.7	37.9
70%	2287.5	3150.0	112.5	450.0	312.5	800.0	25.1	35.8
80%	2275.0	2550.0	112.5	450.0	262.5	750.0	31.0	32.1
100%	2225.0	2950.0	62.5	250.0	212.5	500.0	27.3	27.3

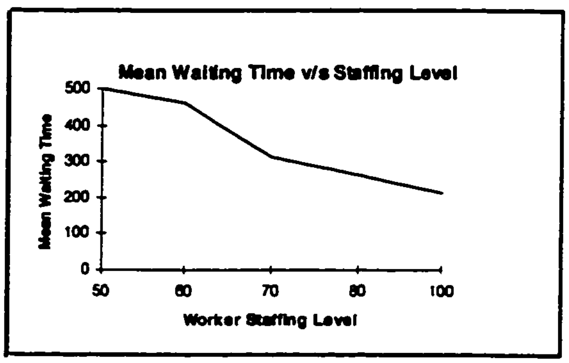
Mach. Util. - Machine Utilization, Work. Util. - Worker Utilization, WT - Waiting Time, FT - Flow Time, T - Tardiness

Table 4.4 Performance Measures verses Worker Staffing Level

<u>**Inventory Related Criteria**</u>



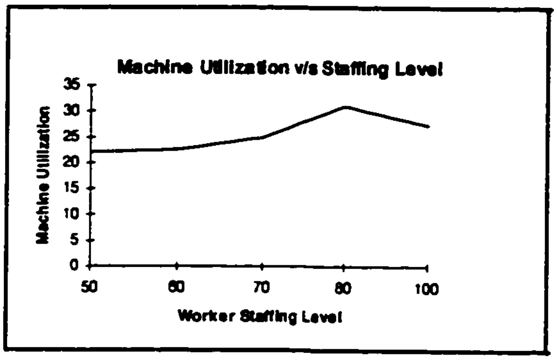
(a)



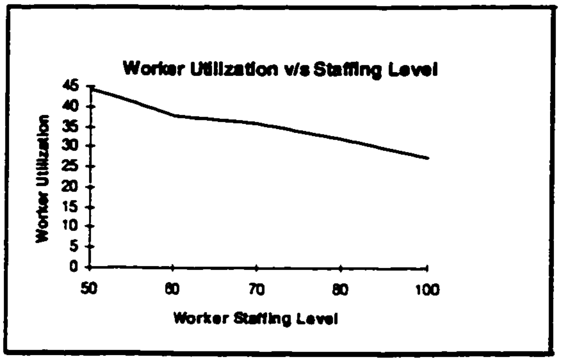
(b)

Figure 4.12 Performance Measures v/s Worker Staffing Level

Resource Utilization Criteria

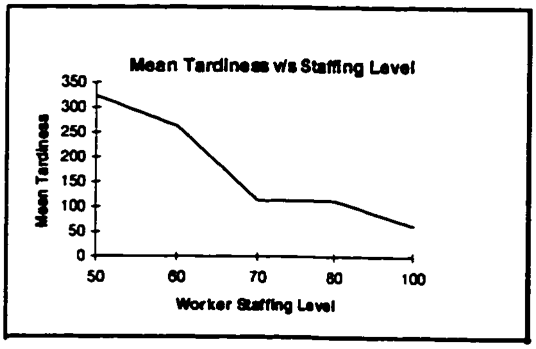


(c)

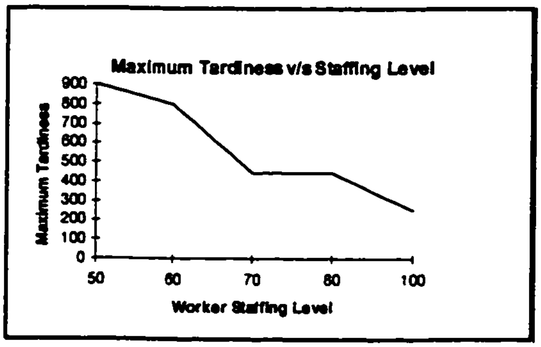


(d)

Due Date Criteria



(e)



(f)

Figure 4.12 (Contd.) Performance Measures v/s Worker Staffing Level

Staffing Level Increase	Percentage Increase in Direct Worker Cost	Percentage Improvement in Mean Flow Time
50% - 60%	20%	1.5%
60% - 70%	16.66%	6.6%
70% - 80%	14%	0.05%
70% - 100%	43%	2.73%

Table 4.5 Direct Worker Cost verses Mean Flow Time

Further considering the machine-constrained shop where equipment (machines) are fully staffed (100% worker staffing), and the workforce is inflexible, did not cause much improvement in mean flow time and other performance measures which can justify the return on investment by increasing the direct worker cost. From table 4.5, it can be observed that increasing staffing level from 70% with flexibility level of 2 to 100% with flexibility level of 1 resulted in increasing the cost of direct workers by 43% with an improvement in mean flow time of 2.8%. The gantt chart for problem A1 is as shown in figure 4.13.

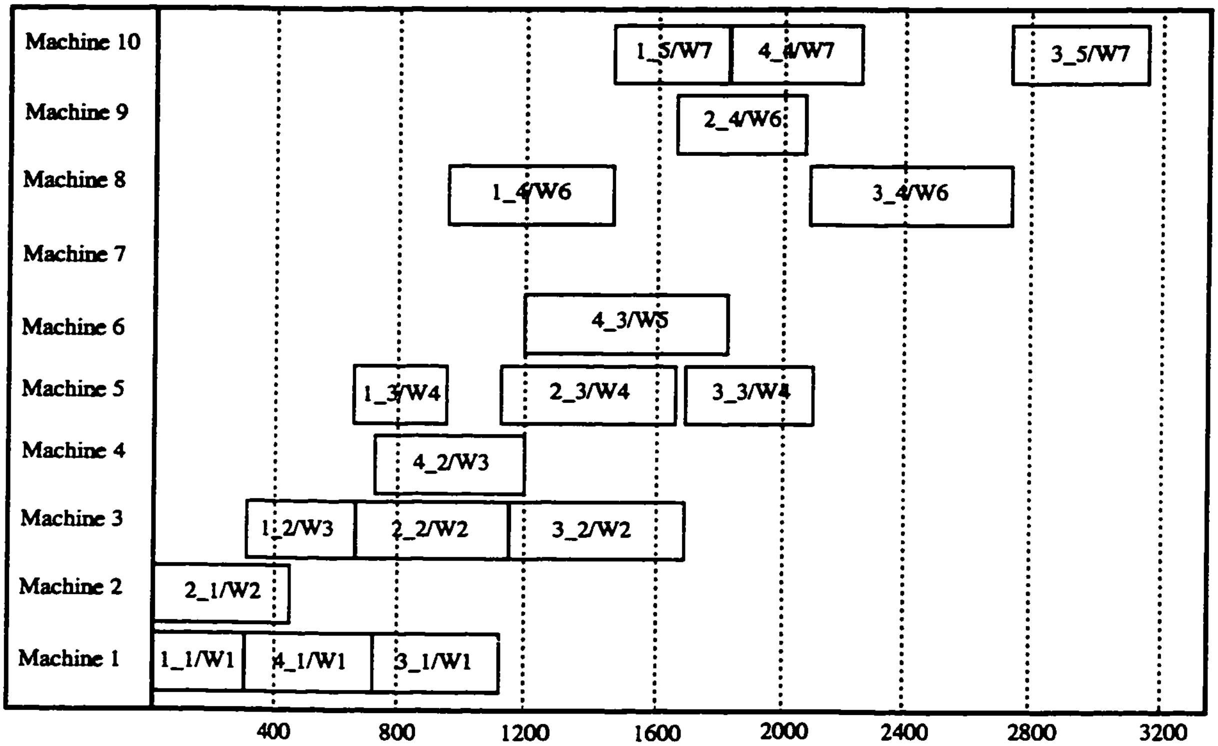


Figure 4.13 Gantt Chart for Problem A1

The second problem (B1) was extracted from Nasr and Elsayed (1990) with some necessary additions such as constraining the problem by worker resources. The problem (B1) consists of four jobs and six machines and each job has three operations to be performed in a given sequence. Each operation has choice of alternate machines on which processing can take place. Nasr and Elsayed (1990) solved the problem using a branch and bound optimization algorithm. The best found schedule using this algorithm for mean flow time was 61.25 time units. Jain (1995) solved the same problem using genetic algorithms for machine constrained shop and the best schedule found for mean flow time was 58.75 time units. Similar results can be observed from table 4.6 for our proposed algorithm. It can be observed from experimentation results (table 4.6) that when only machine is scarce resource, the mean flow time is 58.75 time units. In this case the machine remains idle when there are no jobs for processing. By considering machine and worker both as the scarce resource, the mean flow time observed for worker staffing level of 50% (flexibility of 2) was 63.75 time units. There is a difference in value of mean flow time because now the shop is constrained by both worker and machine resource. In this case machine remains idle when there are no jobs for processing or when there is no worker available to operate the machine. But when the staffing level was 70% (flexibility level of 2), the value of mean flow time was 58.75 time units, which indicates that shop has enough capacity of worker resource to process the jobs. Hence, it can be concluded that when the shop has sufficient number multi skilled workers the optimal staffing level is 70%.

Worker Staffing Level	Mean Flow Time
50%	63.75
67%	58.75
84%	58.75
100%	58.75

Table 4.6 Experimentation Results for Problem (B1)

Once the genetic algorithm parameters have been optimized and optimum level of worker staffing level determined, the next step is the analysis of dispatching rules with respect different performance measures such as mean flow time, mean tardiness, and mean worker utilization. This research has considered the performance of six different dispatching rules and has compared the results of these dispatching rules for two different shop characteristics, i.e. single constrained shop (machine-limited model) and dual constrained shop. Table 4.7 and 4.8 summarize the results of the experiments which compare the performance of six dispatching rules with respect to eight performance measures for a dual resource constrained shop and single resource constrained shop.

Rules	**Mean FT**	**Maxi. FT**	**Mean T**	**Maxi. T**	**Mean WT**	**Maxi. WT**	**Mach. Util.**	**Worker Util.**
SPT	2287.5	2850	112.5	350	262.5	600	31.1%	44.5%
EDD	2325	2750	162.5	350	287.5	600	28.9%	43.1%
LPT	2300	2750	112.5	500	250	650	30.6%	43.5%
EOPNDD	2300	2800	137.5	350	262.5	600	30.7%	43.4%
LSO	2300	2650	112.5	350	250	750	30.8%	42.6%
FCFS	2375.5	2700	175	500	387.5	750	30.0%	41.9%

Table 4.7 Performance Measures v/s Dispatching Rules for Staffing Level 70%

Rules	Mean FT	Maxi. FT	Mean T	Maxi. T	Mean WT	Maxi. WT	Mach. Util.	Worker Util.
SPT	2225	2650	62.5	250	212.5	400	31.3%	31.3%
EDD	2250	2650	62.5	300	200	450	31.3%	31.3%
LPT	2237.5	2500	75	200	175	450	31.7%	31.7%
EOPNDD	2212.5	2500	62.5	250	175	450	32.0%	32.0%
LSO	2250	2550	62.5	250	237.5	500	31.5%	31.5%
FCFS	2350	2750	187.5	400	225	650	29.8%	29.8%

Table 4.8 Performance Measures v/s Dispatching Rules for Staffing Level 100%

The dispatching rule to be employed will be:

(a) Shortest Processing Time (SPT)

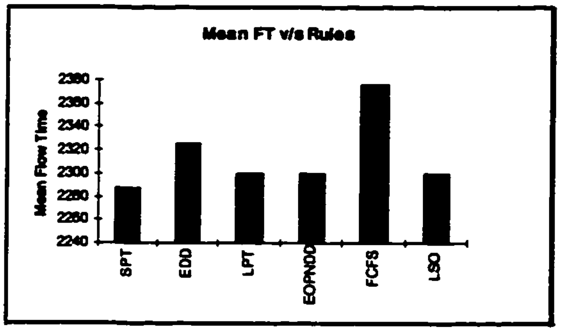
(b) Earliest Operation Due Date (EOPNDD)

(c) Earliest Due Date (EDD)

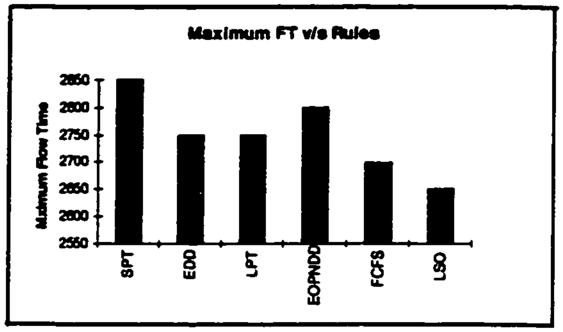
(d) Least Slack Time/Operation (LSO).

(e) First Come First Serve (FCFS)

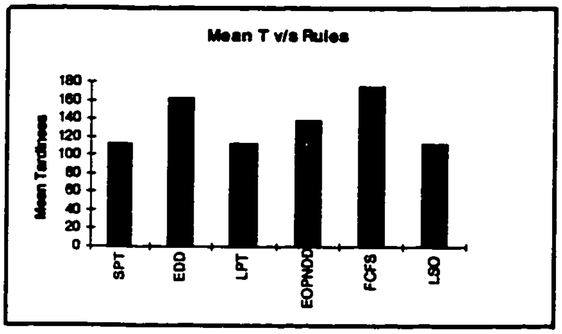
(f) Longest Processing Time (LPT)



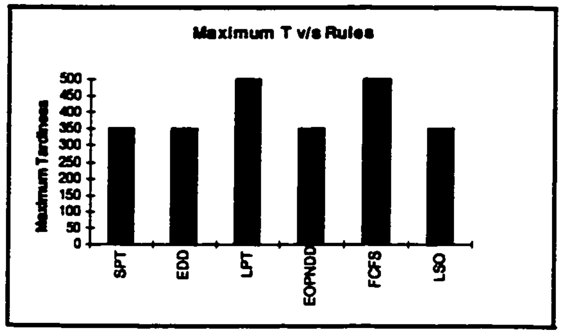
(a)



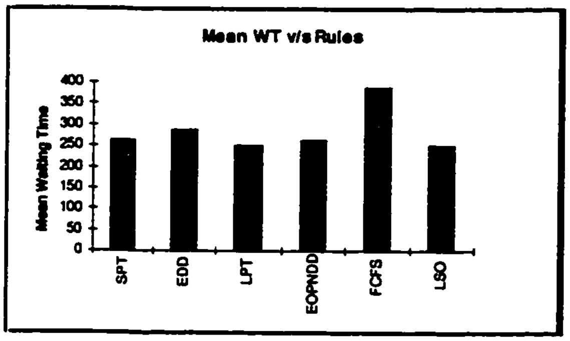
(b)



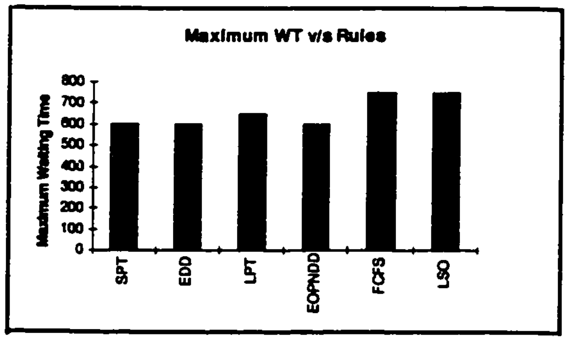
(c)



(d)

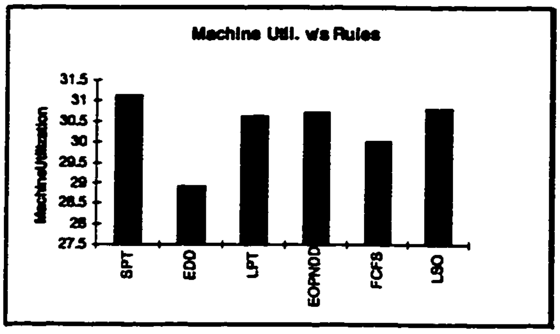


(e)

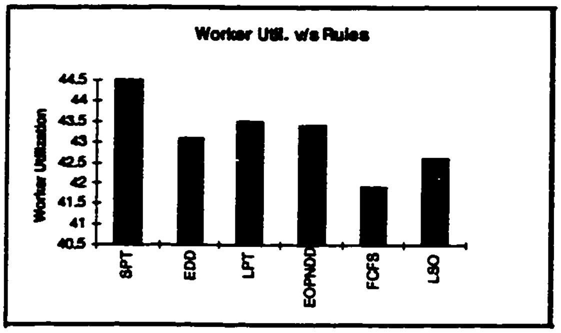


(f)

Figure 4.14 Performance Measures v/s Rules for Worker Staffing level 70%



(g)

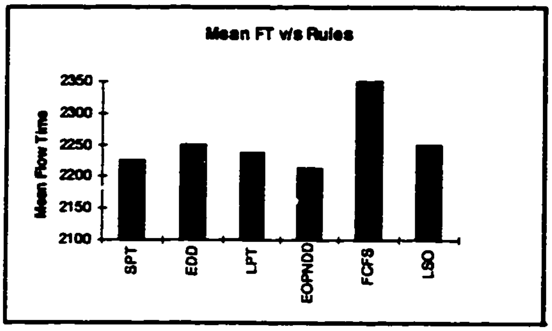


(h)

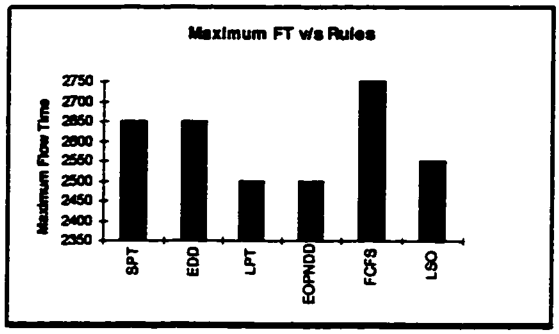
Figure 4.14 Performance Measures v/s Rules for Worker Staffing Level 70%

For a dual resource constrained shop and worker staffing level equal to 70% with flexibility of 2, and performance measure is to minimize the mean flow time, the SPT rule performed well over other rules and was clear winner. But for single constrained shop (machine-limited model) and performance measure to minimize mean flow time, EOPNDD rule was found to be the best rule over all other dispatching rules considered. From figure 4.14a and figure 4.15a, the FCFS rule was the worst performing rules for mean flow time, for both single constrained and dual constrained shop.

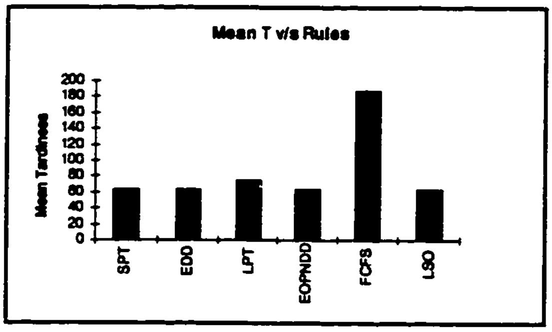
When performance measure is to minimize maximum mean flow time, LSO rule performed the best and SPT was worst performing rule (figure 4.14b) for a dual resource constrained shop. For single resource constrained shop EOPNDD rule was the best performing rule and FCFS was the worst performing rule (figure 4.15b).



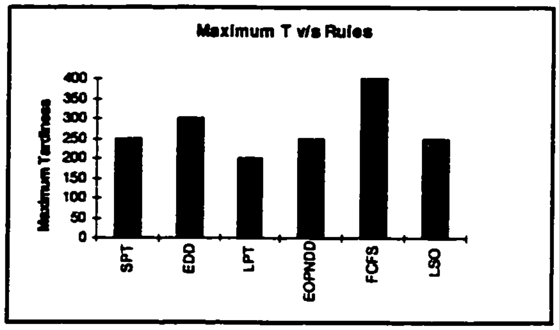
(a)



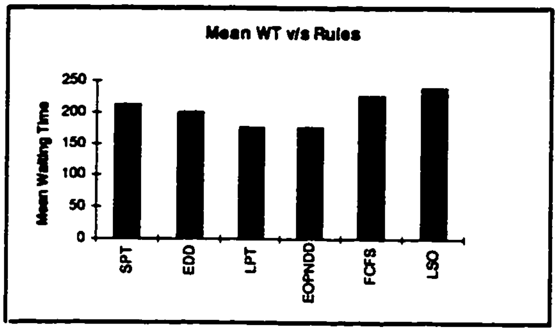
(b)



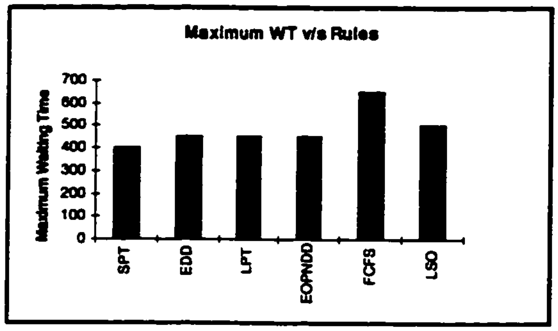
(c)



(d)

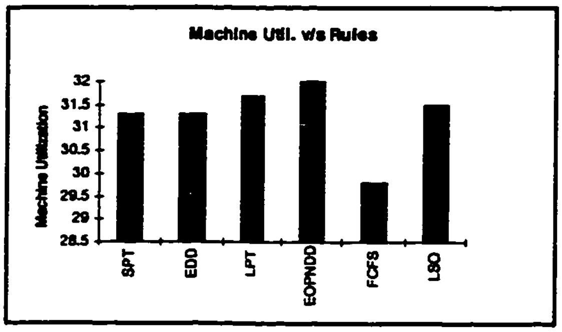


(e)

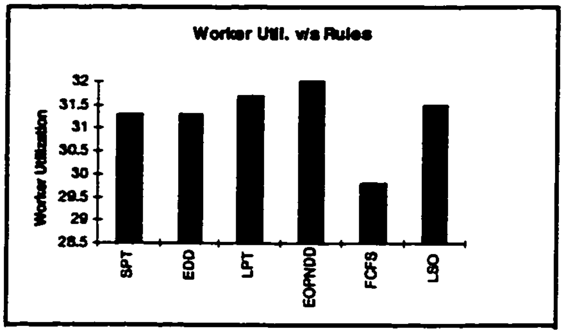


(f)

Figure 4.15 Performance Measures v/s Rules for Worker Staffing Level 100%



(f)



(g)

Figure 4.15 Performance Measures v/s Rules for Worker Staffing Level 100% (Cont)

When mean tardiness was used as the measure SPT, LPT and LSO rule performed well compared to other rules for dual resource constrained, but for single resource constrained EDD, EOPNDD, SPT rules performed well compared to FCFS which was worst performing rules (figure 4.15c). For minimizing maximum tardiness LPT rule which performed well for mean tardiness was the worst performing rule for maximum tardiness for dual resource constrained shop, but for single resource constrained shop LPT was the best performing rule compared to SPT, EOPNDD, EDD. When mean waiting time is used as the measure LPT was found to be a better performing rule compared to other rules for both dual resource and single resource constrained shop. For minimizing maximum waiting time SPT rule found to be a better performing rule for both dual resource and single resource constrained shop. For resource utilization as the performance measure SPT was found to be a better performing rule for dual resurrect utilization, while EOPNDD rule performed better for single resource constrained shop.

The use of a particular scheduling rule is very much dependent on the chosen performance measure (Blackstone et.al. 1982). From figure 4.14, for dual resource constrained shop no rule has been a clear winner for all the performance measures considered, similar conclusions can be drawn for single resource constrained shop. For the example considered here, SPT rule performed fairly well for a dual resource constrained shop, while EOPNDD rule performed well for a single resource constrained shop.

4.9 Summary

We used a new solution approach to solve a scheduling problem when shop capacity is constrained both by worker and machine. For research related to Dual Resource Constrained System, it is generally assumed that the processing time of a task is equal on all machines within one workcell. However, in actual systems the processing time may vary, this short coming is addressed here, where variable processing times have been considered.

In this research, we have compared the effects of dispatching rules on various performance measures for both single resource constrained shop and dual resource constrained shop. We have also considered different staffing levels and determined optimal staffing level.

CHAPTER V

SYSTEM PROTOTYPE

5.1 System Overview

The algorithm and procedure discussed in previous chapters are implemented into a simple software structure. The system is implemented on SUN Work station running in an UNIX environment. The basic steps of the developed system can be represented as follows:

(1) Data entry, i.e., enter all given input data regarding job orders and the manufacturing system.

(2) Enter genetic algorithms parameters

(3) Enter the performance measure.

(4) Enter the dispatching rule.

(5) Perform the optimization of assignment of dual resources using genetic algorithms

(6) Test the assignment of dual resources using the dispatching rule.

(7) Specify 'N', the number of best schedules to be saved.

(8) Output in the form of performance measures values and individual machine and worker utilizations for all 'N" schedules.

(9) Output in the form of a schedule which lists start and completion times of all the job orders.

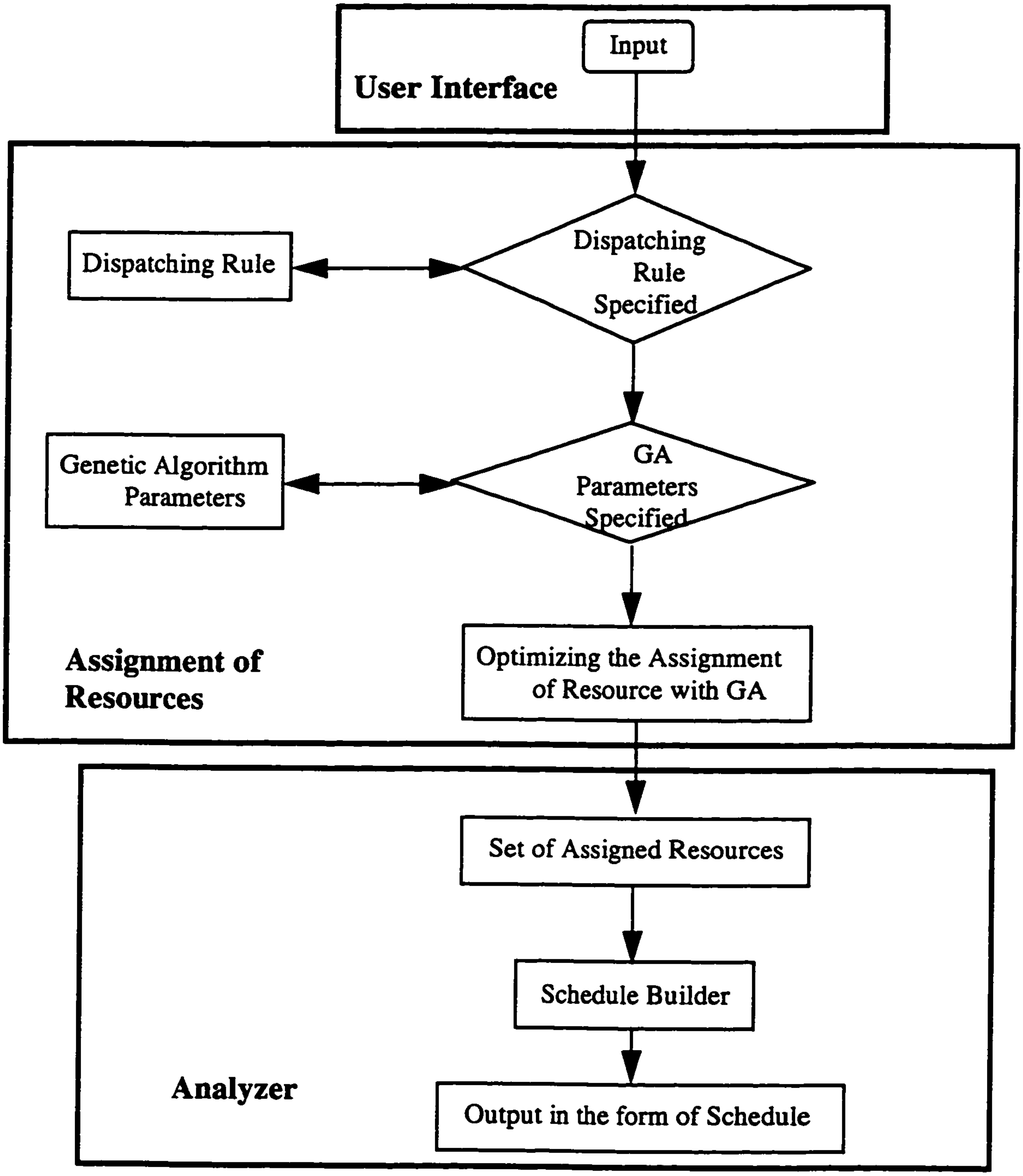


Figure 5.1 Flow Chart of the System

The overall structure of the system is given in figure 5.1. The system consists of three modules, namely user interface, module which makes assignment of resources, and analyzer which provides the start time and completion time of each task with the assigned machine and worker resources. The input requirements are provided through user interface. The second module makes assignment of the available resources based on the input and output requirements. The output from this module is the list of resources to be assigned to various tasks, which are then analyzed by the third module the analyzer. The output from the analyzer will be a schedule which will list start and completion times of all the job orders, and also a detailed information of all tasks in form of which task has to be performed on which machine and by which worker. The sample output from the analyzer is given in Appendix C. The performance criteria to be employed for the system will be Mean Flow Time, but the analyzer lists the values of other performances measures as mentioned below :

(1) Makespan

(2) Mean Tardiness

(3) Mean Waiting Time

(4) Maximum Tardiness

(5) Maximum Waiting Time

(6) Average Machine Utilization

(7) Average Worker Utilization

The dispatching rule to be employed will be specified by the user.

(a) Shortest Processing Time (SPT)

(b) Earliest Operation Due Date (EOPNDD)

(c) Earliest Due Date (EDD)

(d) Least Slack Time/Operation (LSO).

(e) First Come First Serve (FCFS)

(f) Longest Processing Time (LPT)

5.2 User Interface

The user interacts with the system through the user interface. The following are the input requirements to be fed into the system:

(a) Number of Orders to be Scheduled

(b) Number of Machines in the System

(c) Number of Labors in the System

(d) Demand or Order Size of each Order

(e) Number of Operations in each Job

(f) Operation - Machine Requirement

(g) Machine - Worker Information

(h) Processing Time of each Operation

(i) Setup Time of each Operation

(j) Worker Efficiency Information

(k) Due Date of each Order

(l) Due Date Tightness Factor

(m) Performance Criteria

The information entered is then saved in the respective files which are then used by module 2 for optimization of assignment of resources.

5.3 Assignment of Resources

Once all the information has been entered in module 1, then module 2 takes over. In this module the dispatching rule to be used has to be specified. After specifying the dispatching rule, one has to specify the genetic algorithms parameters like

(a) Population Size (PSIZE) (50 to 200)

(b) Selection Bias (SBIAS) (1.0 to 2.0)

(c) Adaptive Mutation Rate (AMUT) (0.0 to 1.0)

(d) Number of Generations (NGEN) (100 to 2000)

The suggested values of the genetic algorithms parameters fall with in the range of the implemented parameters. Once the dispatching rule and genetic algorithms parameters are specified, then the optimization of assignment of dual resources are performed using genetic algorithms.

5.4 Analyzer

The output from module 2 will be list of resources to be assigned to various tasks. The schedule builder analyzes the assigned resources based on the performance criteria and dispatching rule specified. The genetic algorithm works on a population of encoded solutions to the scheduling problem, not on the schedule itself. Since, the representation does not directly represent a schedule, a conversion from chromosome representation to a conventional production schedule has to be performed by a schedule builder. The schedule builder guarantees the feasibility and consistency of the produced schedules. The complexity of the schedule builder depends upon the amount of information encoded in the chromosome representation.

The schedule builder has three basic functions to be performed as follows:

(a) It searches locally for information not supplied by the chromosomes.

(b) It enforces constraints such as

i) a limit on the number of machines that can process a job simultaneously.

ii) a limit on the number of machines a worker can operate simultaneously.

(c) It builds the actual schedules for all of the shops job orders.

The output from the analyzer will be schedule which will list start and completion times of all the job orders.

5.5 Special Features

The special features of the system includes:

(i) Modify existing data, i.e. after data entry session or some other time in future, if the user wishes to modify or change some existing data for running a different experiment it is made possible through the individual data files created during the first interactive session. the very first information that is required of the user is whether the user wants to enter an entirely new set of data or modify existing data. This is shown in figure 5.2.

```
Do you wish to enter new data or modify existing data ?

1. Enter entirely new scheduling data.
2. Modify existing scheduling data.

Enter your choice :-> 2

You can change the following information in the existing files:

1. Modify the Demand of the order
2. Modify the Machine-Labour Requirement of a particular task.
3. Modify the Setup time of a particular task.
4. Modify the processing time of a particular task.
5. Modify Efficiency of Labour.
6. Add a new order.
7. Add a new machine or labour.

Enter your choice :-> 3
```

Figure 5.2 Sample for data modification

(ii) Correct data at any time during data entry, i.e., during the data entry session, if the user finds that an incorrect data has been entered, it is made possible to rectify the incorrect value. After each data entry session, the system responds with a list of values that have been entered and a question about whether the user wishes to change any value. If the user finds that some of the values entered are incorrect, he or she can do by answering yes at this time.

(iii) The system includes a help module which the user can access at any time by entering "?" during the data entry session.

CHAPTER VI

CONCLUSIONS

6.1 Overview

Previous work in manufacturing scheduling is limited to machine constrained job shops and ignored the potential scarcity of worker as an other limiting resource. The research related to scheduling in a dual resource constrained environment to date is also limited to analytical techniques and simulation techniques. The main criticism of using analytical techniques is that for a large sized problems no optimal solution can be found in reasonable amount of time. Another drawback of this approach is that because of the nature of the formulation, it is difficult to model all the details of the problem. There are several drawbacks of using the simulation approach. Simulation can potentially be expensive and time consuming to develop, debug and run. The accuracy of any simulation model is limited by the judgement and skill of the programmer. The other drawback is the large amount of time this approach takes to reach to an optimal solution because of its experimental nature. The basic problem of using heuristics for assignment of resources is that they usually only handle a single measure of performance, and their locally greedy strategies ignore the possibility for the global optimization.

The genetic algorithms have been successfully implemented for solving different scheduling problem for a single resource constrained job shop. It has been observed that genetic algorithms are very effective and efficient in computation time for generating the optimal or near optimal schedules compared to other methods. They are more suitable for optimization problems, not only from an economic point of view, but also from a practical application point of view. The applications of genetic algorithm suffers from a major drawback which is that they have considered only machines as the scarce resource and neglected other resources like workers, tools, pallets, jigs and fixtures.

6.2 Research Summary

This research effort explored the feasibility of using genetic algorithms to solve the production scheduling problems in dual resource constrained systems. A prototype system was developed which can handle dual resources. This prototype system consists of user interface for inputting scheduling data to the system, optimizer for efficient assignment of worker and machine resources to each task, and the analyzer which gives the output in the form of schedule indicating the starting time and completion time of each task with best assigned machine and worker resources. The system developed can be used by the shop floor manager to find optimal staffing level which can be viewed as a basic system design decision as far as the initial hiring is concerned. This system also helps in making short-range control decisions with respect to allocation of available resources (machines and workers) to task and determining start time and completion time of each task. The developed prototype is illustrated with two numerical examples.

6.3 Conclusions

The following conclusions are developed based on the numerical results generated by the conducted experiments.

(1) It is observed that for scheduling discrete manufacturing open shops with dual resources, where each worker has skills to operate two machines (flexibility level 2), and where the operator busy time is considered the same as the machine busy time, the 70% staffing level was found to be optimal. Under these conditions, increasing the staffing level beyond 70% has a minimal gain or the return on investment does not justify the increase in worker staffing level beyond 70%. These results are similar to the results provided by Felan III et.al. (1993). Also increasing the staffing level resulted in increase of direct cost and hence the decision to adopt the strategy of staffing thereby improving the primary performance measures such as mean flow time, mean tardiness and mean waiting time must be balanced by managers against the resulting increase in direct cost.

(2) There is difference in performance measure values when dual resource constrained system with worker staffing level below 70% (flexibility of 2) is compared with single resource constrained system (machine-limited model). This is because, for a machine-limited model machines are idle only when there are no jobs for processing. But for dual

resource constrained system machines remain idle when there are no jobs for processing or when there is no worker available for operating the machine.

(3) From the available literature, it was observed that heuristic rules are applied for solving DRC scheduling problems, and has been limited to a single performance measures. Moreover, none of the research has directly compared the effects of dispatching rules for a dual resource constrained shop and a single resource constrained shop for different performance measures. In this research six dispatching rules, in combination with eight performance measures, have been used while scheduling with genetic algorithms. Given a performance measure and a scheduling rule, the genetic algorithm approach determines the best schedule. It has been observed that the rule which works best for a single resource constrained shop is not necessarily be the best rule for a dual resource constrained system. Further, it is observed that the choice of a particular dispatching rule is dependent on the performance criterion considered and the characteristics of manufacturing systems. This is illustrated with a numerical example. For the example considered here SPT performed fairly well for a dual resource constrained shop, while EOPNDD rule performed well for a single resource constrained shop.

(4) The available literature for studies pertaining to worker constraints has considered the need for a particular type of worker to be only machine dependent. The genetic algorithm approach, developed in this research, considered that the need for a particular type of worker is both machine and job dependent. This approach makes it possible for the proposed genetic algorithm to be adapted to different manufacturing situations with minor modifications pertinent to the given system. For example, the same algorithm can be

applied for the situations where the shop is constrained by machines and tools. Moreover, the same algorithm can be applied for a situation where the workers have varying efficiencies on different machines.

(5) The genetic algorithms approach is much simpler to apply than other optimization and scheduling approaches to solve a DRC scheduling problems. It avoids the use of complex worker assignment rules, which is required when using simulation techniques.

(6) The developed genetic algorithm has been successfully applied in determining the optimal work force required in the system, which is viewed as a basic system design decision as far as the initial hiring is concerned. This algorithm also helps in making short-range control decisions with respect to allocation of a fixed work force, determining start and completion time of each task with best assigned machine and worker resources.

6.4 Recommendations for Future Research

There are some general directions in which the work described in this research may be extended. The following areas are recommended for future research:

(1) In this research, we considered that the shop capacity is constrained by workers and machines, this can be extended further taking into consideration that the shop is constrained by machine, worker and auxiliary resources like special jigs and fixtures, tools, pallets.

(2) In studying the scheduling problem, an assumption was made that there is no loss of productive time when workers are transferred from one machine to another. In general, this is not the case, scheduling of worker and machines with worker transfer delay is a potential area of future research.

(3) In this research, it was assumed that transportation time between facilities is negligible, which may not be the case in some situation. Scheduling of material handling equipment (robots, AGVs etc.) in conjunction with parts and machines is a potential area of future research.

(4) Another relevant issue is the scheduling or dispatching rules used to control the production. Further research is needed to study the impact of various dispatching rules other than those used in this research for a dual resource constrained shop.

(5) In this research, optimal staffing level was determined considering the shop consisting of homogeneous resources. It would be an interesting study to investigate it for a shop comprising of heterogeneous workers.

(6) For a machine constrained shop, it has been observed that the performance of the manufacturing system improves significantly using batch splitting. It would be an interesting study to investigate the performance of batch splitting for a dual resource constrained shop.

(7) Another topic for future research is to use other search techniques like tabu search, simulated annealing and comparing the results with genetic algorithms.

(8) In this research we used reduced surrogated crossover operator, while a number of alternatives are available in the literature. Further research can be carried out to investigate the performance of other alternative crossovers for the dual resource constrained scheduling problems.

(9) In this research worker is assumed to be required for the entire duration of operation on a machine, but this may not be the case every time. Future research is required to consider that worker is occupied only for a part of total operation time (i.e. during loading and unloading time) and is free for the rest of the time.

REFERENCES

(1) Allen, M., (1963), "The Efficient Utilization of Labour Under Conditions of Fluctuating Demand." In J. Muth and G. Thompson (Eds.).

(2) Bagchi, S., Uckun, S., Miyabe, Y., and Kawamura, K., (1991), *"Exploring Problems-Specific Recombination Operators for Job Shop Scheduling."* Proc. of the Fourth Int. Conf. on Genetic Algorithms, San Diego,CA, 10-17.

(3) Baker, K. R., (1974), Introduction to Sequencing and Scheduling, John Wiley and Sons Inc. New York.

(4) Beasley, D., Bull, D. R., and Martin, R. R., (1993a), *"Overview of Genetic Algorithms: Part1, Fundamentals."* University Computing, Vol. 15, No. 2, 58-69.

(5) Biegal, J., and Davern, J., (1990), *"Genetic Algorithms and Job Shop Scheduling." Computers Indust. Engg.*, Vol. 19, 81-91.

(6) Bobrowski, P. M., and Park, P. S., (1993), *"An Evaluation of Labour Assignment rules when workers are not perfectly interchangeable."* Operations Management, Vol. 11, 257-268.

(7) Chen, C. L., Neppali, R. V., and Tanaka, H., (1996), *"Genetic Algorithms Applied to the Continuous Flow shop Problem." Computers Indust. Engg.*, Vol. 30, No. 4, 919-929.

(8) Cleveland, G. A.., and Smith, S. F., (1989), *"Using Genetic Algorithms to Schedule Flow Shop Releases."* Proc. of the Third International Conf. on Genetic Algorithms, 160-169.

(9) Conway, R. W., Maxwell, W. L., and Miller, L. W., (1967), Theory of Scheduling, Addison - Wesley, Reading, MA.

(10) Davis, L., (1985), *"Job Shop Scheduling with Genetic Algorithms."* Proceedings of an International Conf. on Genetic Algorithms and their applications, Carnegie-Mellon University, Pittsburgh, PA, 136-140.

(11) Davis, L., (1991), Handbook of Genetic Algorithms, Van Nostrand Reinhold Publishers, New York.

(12) Falkenauer, E., and Bouffouix, S., (1991), *"A Genetic Algorithm for Job Shop."* Proc. of the 1991 IEEE International Conf. on Robotics and Automation, 824-829.

(13) Felan III, J. T., Fry, T. D., and Philipoom, P. R., (1993), *"Labour Flexibility and Staffing levels in a dual resource constrained job shop."* International Journal of Production Research, Vol. 31, No. 10, 2487-2506.

(14) Fryer, J. S., (1974), *"Labour Flexibility in Multiechelon Dual Constrained Job Shop."* Management Science, Vol. 20, No. 7, 1073-1080.

(15) Fryer, J. S., (1975), *"Effects of Shop Size and Labour Flexibility in Labour Production System."* Management Science, Vol. 21, No. 5, 507-515.

(16) Fryer, J. S., (1976), *"Organizational segmentation and labor transfer policies in labor and machine limited Production Systems."* Decision Science, Vol. 7, 725-738.

(17) Fox, B. R., and McMohan, M. B., (1991), *"Genetic Operators for Sequencing Problems."* Foundations of Genetic Algorithms, Morgan Kaufmann, 284-300.

(18) Fox, M. S., and Smith, S. F., (1984), *"ISIS: A Knowledge-Based System for Factory Scheduling."* Expert Systems, Vol. 1, No. 1, 25-49.

(19) Garey, M. R., and Johnson, D. S., (1979), Computers and Intractability, Freeman and Company, San Francisco, CA.

(20) Goldberg, D. E., (1989), Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company Inc., Reading, MA.

(21) Graves, S. C., (1981), "*A Review of Production Scheduling*." Operations Research, Vol. 29, No. 4, 646-675.

(22) Grefenstette, J. J., (ed.) (1985), Proceedings of the First International Conference on Genetic Algorithms.

(23) Gunther, R. E., (1979), *"Server Transfer Delays in a Dual Resource Constrained Paralleled Queuing System."* Management Science, Vol. 25, No. 12, 1245-1257.

(24) Gunther, R. E., (1981), *"Dual Resource Parallel Queues with Server Transfer and Information Access Delays."* Decision Science, Vol. 12, No. 1, 97-111.

(25) Gupta, M. C., Gupta, Y. P., and Evans, G. W., (1993), *"Operations Planning and Scheduling problem in advance manufacturing systems."* International Journal of Production Research, Vol. 31, No. 4, 869-900.

(26) Gupta, M. C., Gupta, Y. P., and Kumar, A., (1993), *"Minimizing flow time variance in a single machine system using genetic algorithms."* European Journal of Operational Research, Vol. 70, 289-303.

(27) Hogg, G. L., Phillips, D. T., and Maggard, M. J., (1977), *"Parallel-Channel dual resource constrained queuing systems with heterogeneous resources."* AIIE Transactions, Vol. 9, No. 4, 352-362.

(28) Holland, J. H., (1975), Adaptation in Natural and Artificial Systems, University of Michigan, Ann Arbor, MI.

(29) Jain, A. K., (1995), *"An Integrated Scheduling Approach for Discrete Manufacturing Systems."* Ph.D. Thesis, McMaster University, Ontario, Canada.

(30) Jain, A. K., and ElMaraghy, H. A., (1994), *"The use of Batch Sizing to Improve Flow and Waiting Times in FMS."* 4th Int. Conference on Computer Integrated Manufacturing and Automation Technology, Rensselaer Polytechnic Institute, Troy, NY, 10-12 October, 403-408.

(31) Jain, A. K. and ElMaraghy, H. A., (1994), *"Manufacturing Scheduling using Genetic Algorithms."* CSME'94 Forum, Montreal, QC, June 20-22, 712-727.

(32) Jain, A. K., and ElMaraghy, H. A., (1994), *"Dynamic Scheduling in Flexible Manufacturing Systems."* 10th ISPE/IFAC Int. Conf. on CAD/CAM, Robotics and Factories of the Future (CARS & FOF'94), Ottawa, ONT, August 21-24, 119-125.

(33) Jain, A. K., and ElMaraghy, H. A., (1995), *"A Genetic Algorithms Approach to Production Planning for Batch Manufacturing Systems."* 27th CIRP Int. Seminar on Manufacturing Systems, Ann Arbor, MI, 21-23 May, 347-356.

(34) Jain, A. K., and ElMaraghy, H. A., (1997), *"Production Scheduling/Rescheduling in flexible manufacturing."* International Journal of Production Research, Vol. 35, No. 1, 281-309.

(35) Jain, A.K., and ElMaraghy, H. A., (1997), *"Single Process Plan Scheduling with genetic algorithms."* Production Planning & Control, Vol. 8, No. 4, 363-376.

(36) Khuri, S., Bach, T., and Heitkotter, J., (1994), *"An Evolutionary Approach to Combinatorial Optimization Problems."* Proceedings of Computer Science Conf., ACM 22nd Annual Conf., Phoenix Arizona, March 8-10, 66-73.

(37) Malhotra, M. K., and Kher, H. V., (1994). *"An Evaluation of worker assignment policies in dual resource constrained job shops with heterogeneous resources and worker transfers delays."* International Journal of Production Research, Vol. 32, No. 5, 1087-1103.

(38) Michalewicz, Z. (1992), Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag Publishers, New York.

(39) Murata, T., Ishibuchi, H., and Tanaka, H., (1996), *"Multi-Objective Genetic Algorithms and its Application to flowshop scheduling."* Computers Indust. Engg., Vol. 30, No. 4, 1061-1071.

(40) Nakano, R., and Yamada, T., (1991), *"Conventional Genetic Algorithm for Job Shop."* Proc. of the Fourth International Conf. on Genetic Algorithms, San Mateo, CA, 4774-4779.

(41) Nelson, R. T., (1970), *"A Simulation of Labour Efficiency and Central Assignment in a Production Model."* Management Science, Vol. 17, No. 2, 97-106.

(42) Norman, B. A., (1995), *"The Random Keys Genetic Algorithm for Complex Scheduling Problems."* Ph.D. Thesis, University of Michigan, Michigan.

(43) Park, P. S., (1991), *"The Examination of worker cross-training in a dual resource constrained job shop."* European Journal of Operations Research, Vol. 51, 291-299.

(44) Rochette, R., and Sadowski, R. P., (1976), *"A statistical comparison of the performance of simple dispatching rules for a particular set of job shops."* International Journal of Production Research, Vol. 15, No. 1, 63-75.

(45) Treleven, M. D., and Elvers, D. A., (1985), *"An Investigation of Labour Assignment rules in a Dual Resource Constrained job-shop."* Operations Management, Vol. 6, No. 1, 51-68.

(46) Treleven, M. D., (1987), *"The Timing of Labour Transfer in Dual Resource Constrained System: 'Push' v/s 'Pull' Rules."* Decision Sciences, Vol. 18, No. 1, 73-88.

(47) Treleven, M. D., (1989), *"A review of the dual resource constrained system research."* IIE Transactions, Vol. 21, 279-287.

(48) Verbraeck, A. (1991), *"Developing an Adaptive Scheduling Support Environment."* Ph.D. Thesis, Delft University, Delft, Netherlands.

(49) Weeks, J. K., and Fryer, J. S., (1976), *"A Simulation Study of Operating policies in hypothetical dual-constrained job shop."* Management Science, Vol. 22, No. 12, 1362-1371.

(50) Whitley, D., and Kauth, J., (1988), *"GENITOR: a Different Genetic Algorithm."* Proceedings of the Rocky Mountain Conference on Artificial Intelligence, Denver, CO, 118-130.

APPENDIX A

NUMERICAL EXAMPLE A1

Data set for Example A1

Job	Task	Machine Resources									
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Inner Race Bearing Type 6306	1-1	5	7	-	-	-	-	-	-	-	-
	1-2	-	-	8	10	-	-	-	-	-	-
	1-3	-	-	-	-	5	8	-	-	-	-
	1-4	-	-	-	-	-	-	10	11	-	-
	1-5	-	-	-	-	-	-	-	-	8	7
Outer Race Bearing Type 6306	2-1	8	9	-	-	-	-	-	-	-	-
	2-2	-	-	10	13	-	-	-	-	-	-
	2-3	-	-	-	-	10	12	-	-	-	-
	2-4	-	-	-	-	-	-	-	-	8	7
Inner Race Bearing Type 6308	3-1	7	9	-	-	-	-	-	-	-	-
	3-2	-	-	11	12	-	-	-	-	-	-
	3-3	-	-	-	-	8	10	-	-	-	-
	3-4	-	-	-	-	-	-	14	13	-	-
	3-5	-	-	-	-	-	-	-	-	9	8
Outer Race Bearing Type 6308	4-1	10	13	-	-	-	-	-	-	-	-
	4-2	-	-	11	9	-	-	-	-	-	-
	4-3	-	-	-	-	10	12	-	-	-	-
	4-4	-	-	-	-	-	-	-	-	8	7

Figure A1.1 Task - Processing Time Information for Example A1

Worker - Machine Information for Problem A1

Worker	Machine Resources									
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
W1	1.0	1.0	1.0	-	-	-	-	-	-	-
W2	1.0	-	1.0	1.0	-	-	-	-	-	-
W3	-	-	-	-	1.0	1.0	-	-	-	-
W4	-	-	-	-	-	-	1.0	1.0	-	-
W5	-	-	-	-	-	-	-	-	1.0	1.0

Figure A1.2a Worker Staffing Level 50%

Worker	Machine Resources									
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
W1	1.0	-	1.0	-	-	-	-	-	-	-
W2	-	1.0	-	1.0	-	-	-	-	-	-
W3	-	-	-	-	1.0	-	1.0	-	-	-
W4	-	-	-	-	-	1.0	-	1.0	-	-
W5	-	-	-	-	-	1.0	-	-	1.0	-
W6	-	-	-	-	1.0	-	-	-	-	1.0

Figure A1.2b Worker Staffing level 60%

Worker	Machine Resources									
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
W1	1.0	1.0	-	-	-	-	-	-	-	-
W2	-	1.0	1.0	-	-	-	-	-	-	-
W3	-	-	1.0	1.0	-	-	-	-	-	-
W4	-	-	-	-	1.0	1.0	-	-	-	-
W5	-	-	-	-	-	1.0	1.0	-	-	-
W6	-	-	-	-	-	-	-	1.0	1.0	-
W7	-	-	-	-	-	-	-	-	1.0	1.0

Figure A1.2c Worker Staffing Level 70%

Worker	Machine Resources									
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
W1	1.0	1.0	-	-	-	-	-	-	-	-
W2	-	1.0	1.0	-	-	-	-	-	-	-
W3	-	-	1.0	1.0	-	-	-	-	-	-
W4	-	-	-	-	1.0	1.0	-	-	-	-
W5	-	-	-	-	-	-	1.0	1.0	-	-
W6	-	-	-	-	-	-	1.0	1.0	-	-
W7	-	-	-	-	-	-	-	-	1.0	1.0
W8	-	-	-	-	-	1.0	-	-	1.0	-

Figure A1.2d Worker Staffing Level 80%

Worker	Machine Resources									
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
W1	1.0	-	-	-	-	-	-	-	-	-
W2	-	1.0	-	-	-	-	-	-	-	-
W3	-	-	1.0	-	-	-	-	-	-	-
W4	-	-	-	1.0	-	-	-	-	-	-
W5	-	-	-	-	1.0	-	-	-	-	-
W6	-	-	-	-	-	1.0	-	-	-	-
W7	-	-	-	-	-	-	1.0	-	-	-
W8	-	-	-	-	-	-	-	1.0	-	-
W9	-	-	-	-	-	-	-	-	1.0	-
W10	-	-	-	-	-	-	-	-	-	1.0

Figure A1.2e Worker Staffing Level 100%

APPENDIX B

NUMERICAL EXAMPLE B1

Data Set for Problem B1

Job Type (J_i)	Operation (O_{ij})	Machine Resources M1	M2	M3	M4	M5	M6
1	11	2	3	4	-	-	-
	12	-	3	-	2	4	-
	13	1	4	5	-	-	-
2	21	3	-	5	-	2	-
	22	4	3	-	-	6	-
	23	-	-	4	-	7	11
3	31	5	6	-	-	-	-
	32	-	4	-	3	5	-
	33	0	-	13	-	9	12
4	41	9	-	7	9	-	-
	42	-	6	-	4	-	5
	43	1	-	3	-	-	3

Figure B1.1 Processing Time Information for Example B1

Worker	Machines					
	M1	M2	M3	M4	M5	M6
W1	1.0	-	1.0	1.0	-	-
W2	1.0	1.0	-	-	1.0	-
W3	-	1.0	1.0	-	-	1.0

Figure B1.2a Worker Staffing Level 50%

Worker	Machines					
	M1	M2	M3	M4	M5	M6
W1	1.0	-	1.0	1.0	-	-
W2	1.0	1.0	-	-	1.0	-
W3	-	1.0	1.0	-	-	1.0
W4	-	-	-	1.0	1.0	10

Figure B1.2b Worker Staffing Level 67%

Worker	Machines					
	M1	M2	M3	M4	M5	M6
W1	1.0	-	1.0	1.0	-	-
W2	1.0	1.0	-	-	1.0	-
W3	-	1.0	1.0	-	-	1.0
W4	-	-	-	1.0	1.0	1.0
W5	-	1.0	1.0	1.0	-	-

Figure B1.2c Worker Staffing Level 84%

Worker	Machines					
	M1	M2	M3	M4	M5	M6
W1	1.0	-	-	-	-	-
W2	-	1.0	-	-	-	-
W3	-	-	1.0	-	-	-
W4	-	-	-	1.0	-	-
W5	-	-	-	-	1.0	-
W6	-	-	-	-	-	1.0

Figure B1.2d Worker Staffing Level 100%

APPENDIX C

SYSTEM OUTPUT

++

PSGA
(Priority Scheduling with Genetic Algorithms)

RESULT SUMMARY

Analyst : Vishvas Patel
Project : Bearing_Problem
Version : 1
Date : Thursday, 07 August 1997

Output File : RESULT-Bearing_problem-1
++

INPUT DATA FOR SCHEDULING

SCHEDULING PARAMETERS :

Number of Orders to be scheduled = 4
Number of Machines in the System = 10
Number of Labours in the System = 7
Due date tightness (DDT) factor used = 1.000000
Waiting time limit in any queue = 0

Orders	1	2	3	4
Number of Operations	5	4	5	4
Order Size	50.00	50.00	50.00	50.00
Order Due date	2200.00	2100.00	2700.00	2200.00

Primary Performance Criteria = Mean flow time
Secondary Performance Criteria =
Dispatching Rule = Shortest Processing Time

++

GENETIC ALGORITHM PARAMETERS :

Population Size = 70
Length of the String = 18

Number of Generations = 500
Selection Bias = 1.100000
Mutation Rate = 0.500000
Status Interval = 50
Final Population File = patel

++

TASK'S SETUP TIME/PROCESSING TIME:

Task	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
1_1	0/5	0/7	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
1_2	0/0	0/0	0/8	0/10	0/0	0/0	0/0	0/0	0/0	0/0
1_3	0/0	0/0	0/0	0/0	0/5	0/8	0/0	0/0	0/0	0/0
1_4	0/0	0/0	0/0	0/0	0/0	0/0	0/10	0/11	0/0	0/0
1_5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/8	0/7
2_1	0/8	0/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
2_2	0/0	0/0	0/10	0/13	0/0	0/0	0/0	0/0	0/0	0/0
2_3	0/0	0/0	0/0	0/0	0/10	0/12	0/0	0/0	0/0	0/0
2_4	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/8	0/7
3_1	0/7	0/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
3_2	0/0	0/0	0/11	0/12	0/0	0/0	0/0	0/0	0/0	0/0
3_3	0/0	0/0	0/0	0/0	0/8	0/10	0/0	0/0	0/0	0/0
3_4	0/0	0/0	0/0	0/0	0/0	0/0	0/14	0/13	0/0	0/0
3_5	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/9	0/8
4_1	0/10	0/13	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
4_2	0/0	0/0	0/11	0/9	0/0	0/0	0/0	0/0	0/0	0/0
4_3	0/0	0/0	0/0	0/0	0/10	0/12	0/0	0/0	0/0	0/0
4_4	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/8	0/7

++

Machine/Labour Information:

Machines	L1	L2	L3	L4	L5	L6	L7
M1	1.00	0.00	0.00	0.00	0.00	0.00	0.00
M2	1.00	1.00	0.00	0.00	0.00	0.00	0.00
M3	0.00	1.00	1.00	0.00	0.00	0.00	0.00
M4	0.00	0.00	1.00	0.00	0.00	0.00	0.00
M5	0.00	0.00	0.00	1.00	0.00	0.00	0.00
M6	0.00	0.00	0.00	1.00	1.00	0.00	0.00
M7	0.00	0.00	0.00	0.00	1.00	0.00	0.00
M8	0.00	0.00	0.00	0.00	0.00	1.00	0.00
M9	0.00	0.00	0.00	0.00	0.00	1.00	1.00
M10	0.00	0.00	0.00	0.00	0.00	0.00	1.00

++

PERFORMANCE MEASURES FOR BEST 10 SCHEDULES:

Schedule Number	Mean Flowtime	Max Flowtime	Mean Tardiness	Max Tardiness	Mean Waittime	Max Waittime	Average Mach.Util	Average Lab.Util
1	2287.50	3150.00	112.50	450.00	312.50	800.00	790.00	1128.57
2	2312.50	3250.00	137.50	550.00	325.00	850.00	795.00	1135.71
3	2325.00	3300.00	150.00	600.00	325.00	850.00	800.00	1142.86
4	2337.50	3200.00	162.50	500.00	375.00	850.00	785.00	1121.43
5	2350.00	3250.00	150.00	550.00	375.00	900.00	790.00	1128.57
6	2362.50	3300.00	187.50	600.00	375.00	850.00	795.00	1135.71
7	2375.00	3250.00	212.50	550.00	362.50	850.00	805.00	1150.00
8	2387.50	3300.00	212.50	600.00	387.50	850.00	800.00	1142.86
9	2400.00	3400.00	212.50	700.00	375.00	1000.00	810.00	1157.14
10	2412.50	2950.00	250.00	750.00	362.50	850.00	820.00	1171.43

INDIVIDUAL MACHINE UTILIZATIONS FOR BEST 10 SCHEDULES:

Schedule Number	Mach1 Util.	Mach2 Util.	Mach3 Util.	Mach4 Util.	Mach5 Util.	Mach6 Util.	Mach7 Util.	Mach8 Util.	Mach9 Util.	Mach10 Util.
1	1100.0	450.0	1450.0	450.0	1150.0	600.0	0.00	1200.0	400.0	1100.0
2	1100.0	450.0	900.0	1050.0	1150.0	600.0	0.00	1200.0	400.0	1100.0
3	1100.0	450.0	900.0	1050.0	1150.0	600.0	0.00	1200.0	850.0	700.0
4	1100.0	450.0	1450.0	450.0	1150.0	600.0	500.0	650.0	400.0	1100.0
5	1100.0	450.0	1450.0	450.0	1150.C	600.0	0.00	1200.0	400.0	1100.0
6	1100.0	450.0	900.0	1050.0	1150.0	600.0	500.0	650.0	850.0	700.0
7	1100.0	450.0	500.0	1550.0	1150.0	600.0	500.0	650.0	800.0	750.0
8	1100.0	450.0	900.0	1050.0	1150.0	600.0	500.0	650.0	1250.0	350.0
9	1100.0	450.0	500.0	1550.0	1150.0	600.0	0.00	1200.0	800.0	750.0
10	600.0	1100.0	500.0	1550.0	1150.0	600.0	500.0	650.0	800.0	750.0

INDIVIDUAL LABOUR UTILIZATIONS FOR BEST 10 SCHEDULES:

Schedule Number	Labour1 Util.	Labour2 Util.	Labour3 Util.	Labour4 Util.	Labour5 Util.	Labour6 Util.	Labour7 Util.
1	1100.00	1500.00	850.00	1150.00	600.00	1600.00	1100.00
2	1100.00	950.00	1450.00	1150.00	600.00	1600.00	1100.00
3	1100.00	950.00	1450.00	1150.00	600.00	2050.00	700.00
4	1100.00	950.00	1400.00	1150.00	1100.00	1050.00	1100.00
5	1100.00	1500.00	850.00	1150.00	600.00	1600.00	1100.00
6	1100.00	950.00	1450.00	1150.00	1100.00	1500.00	700.00
7	1100.00	950.00	1550.00	1150.00	1100.00	1050.00	1150.00
8	1100.00	950.00	1450.00	1150.00	1100.00	1500.00	750.00
9	1100.00	950.00	1550.00	1150.00	600.00	2000.00	750.00
10	1250.00	950.00	1550.00	1150.00	1100.00	1450.00	750.00

++

Schedule Number	Primary Performance
1	2287.50
2	2312.50
3	2325.00
4	2337.50
5	2350.00
6	2362.50
7	2375.00
8	2387.50
9	2400.00
10	2412.50

++

THE BEST SCHEDULE IS : 1

1. Performance Measure Values for schedule 1 are:

Minimum Mean Flow Time is	: 2287.50000
Minimum Maximum Flow Time is	: 3150.00000
Minimum Mean Tardiness is	: 112.50000
Minimum Maximum Tardiness is	: 450.00000
Minimum Mean Waiting Time is	: 312.50000
Minimum Maximum Waiting Time is	: 800.00000
Maximum Mean Machine Utilization is	: 0.25079
Maximum Mean Labour Utilization is	: 0.35828

2. Machine Utilizations for schedule 1 are:

Machine Number	Busy Time	Percent Utilization
1	1100.000	0.34921
2	450.000	0.14286
3	1450.000	0.46032
4	450.000	0.14286
5	1150.000	0.36508
6	600.000	0.19048
7	0.00000	0.00000
8	1200.000	0.38095
9	400.000	0.12698
10	1100.000	0.34921

3. Labour Utilizations for Schedule1 are :

Labour Number	Busy Time	Percent Utilization
1	1100.000	0.34921
2	1500.000	0.47619
3	850.000	0.26984
4	1150.000	0.36508
5	600.000	0.19048
6	1600.000	0.50794
7	1100.000	0.34921

++

FOR THE CURRENT SCHEDULE :

Task	Order Name	Machine	Labour	Start Time	Finish Time	Processing Time
1/1	a	1	1	0.00	250.00	250.00
1/2	a	3	3	250.00	650.00	400.00
1/3	a	5	4	650.00	900.00	250.00
1/4	a	8	6	900.00	1450.00	550.00
1/5	a	10	7	1450.00	1800.00	350.00
2/1	b	2	2	0.00	450.00	450.00
2/2	b	3	2	650.00	1150.00	500.00
2/3	b	5	4	1150.00	1650.00	500.00
2/4	b	9	6	1650.00	2050.00	400.00
3/1	c	1	1	750.00	1100.00	350.00
3/2	c	3	2	1150.00	1700.00	550.00
3/3	c	5	4	1700.00	2100.00	400.00
3/4	c	8	6	2100.00	2750.00	650.00
3/5	c	10	7	2750.00	3150.00	400.00
4/1	d	1	1	250.00	750.00	500.00
4/2	d	4	3	750.00	1200.00	450.00
4/3	d	6	5	1200.00	1800.00	600.00
4/4	d	10	7	1800.00	2150.00	350.00

VITA AUCTORIS

Vishvas Patel was born in 1969 in Baroda, Gujarat, India. He graduated from Senior Secondary School in 1986. From there he went on to M.S. University of Baroda, India, where he obtained his Bachelor in Mechanical Engineering in 1991. He then joined FAG-BEARINGS, LTD. INDIA, in 1992 and worked there for two and half years as a Production Engineer. He then joined CHICAGO PNEUMATIC LTD. INDIA, in 1994 and worked there for one year as a Senior Quality Assurance Engineer. He is currently a candidate for the Master's degree in Industrial and Manufacturing Systems Engineering at the University of Windsor and hopes to graduate in Fall 1997.